

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/108829>

Copyright and reuse:

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



Adapting the Gibbs Sampler

by

Cyril Chimisov

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Statistics

January 2018



Contents

Acknowledgments	iv
Declarations	vi
Abstract	vii
List of Algorithms	viii
Abbreviations	ix
Chapter 1 Introduction	1
1.1 Markov Chain Monte Carlo	1
1.1.1 Metropolis-Hastings framework	2
1.1.2 Gibbs Sampler	5
1.2 Adaptive MCMC	7
1.3 Overview of the thesis and main results	13
1.3.1 Adaptive Gibbs Sampler	14
1.3.2 AirMCMC	15
Chapter 2 Adaptive Gibbs Sampler	18
2.1 RSGS spectral gap for Multivariate Gaussian distribution	18
2.2 Pseudo-spectral gap	22
2.3 Motivating examples	24
2.4 Adapting Gibbs Sampler	26
2.5 Adapting Metropolis-within-Gibbs	34
2.6 Ergodicity of the Adaptive Gibbs Sampler	37
2.7 Simulations	43
2.7.1 Adaptive Random Scan Gibbs Sampler	48
2.7.2 Adaptive Random Walk Metropolis within Adaptive Gibbs	54
2.7.3 Computational cost of the adaptation	56

2.8	Discussion	58
2.9	Proofs of the statements from Chapter 2	59
Chapter 3	AirMCMC	71
3.1	Motivating Examples	71
3.1.1	Adaptive Scaling of Random Walk Metropolis	71
3.1.2	Adaptive Metropolis for high dimensional correlated posteriors	74
3.2	AirMCMC Theory	77
3.2.1	Simultaneous Geometric Ergodicity	81
3.2.2	Local Simultaneous Geometric Ergodicity	82
3.2.3	Simultaneous Polynomial Ergodicity	84
3.2.4	Convergence in distribution	85
3.3	Comparison with available Adaptive MCMC theory	89
3.4	Examples: Air versions of complex AMCMC algorithms	91
3.4.1	Adaptive Random Scan Gibbs Sampler	91
3.4.2	Kernel Adaptive Metropolis-Hastings	92
3.5	Discussion	94
3.6	Proofs for Section 3.2	95
3.6.1	Proof of Theorem 10	101
3.6.2	Proof of Theorem 11	104
3.6.3	Proof of Theorem 12	104
3.6.4	Proof of Theorem 13	110
3.6.5	Proof of Theorem 14	111
3.7	Appendix A	111
3.8	Appendix B	116
Chapter 4	Software package	119
4.1	Overview	119
4.2	Installation	120
4.2.1	Compile	120
4.2.2	After compilation	120
4.3	R-defined target densities	121
4.4	C++-defined target densities	122
4.5	Does the adaptation help?	124
4.6	AMCMC function	128
4.6.1	Starting values	128
4.6.2	Adaptations	128
4.6.3	Blocking	128

4.6.4	Full conditional density specification	129
4.6.5	Gibbs sampling	131
4.6.6	Parallel adaptations	132
4.6.7	AirMCMC	133
4.7	Accessing chain output	133
4.8	Customising template.hpp	133
4.9	Discussion	134
Bibliography		135

Acknowledgments

First, I would like to express my genuine gratitude to Dr Krys Łatuszyński and Prof Gareth Roberts. Without their guidance and patience, it would not have been possible to complete this dissertation. It was a great pleasure to have all the fruitful and encouraging discussions, which have led to the development of ideas described in the present work.

I also owe a great deal of gratitude to Murray Pollock, Christian Robert and Matt Moores, with whom I had numerous conversations, which helped me comprehend a large chunk of the Bayesian world. I would also like to thank every member of the Algorithms, Simulations, and Machine Learning reading groups, and, more generally, everyone at the Department of Statistics for creating the dynamic and stimulating environment.

Special thanks to Daniel, Ellie, Neil, Rodrigo, Te-Anne and my office mates who have shaped my life for the much better.

It would not be possible to stay concentrated and motivated throughout the PhD years without my caring and supportive partner Lana. Last but not the least, I would like to thank my mother, who has given me more than I can ever give back.

I would also like to acknowledge the Engineering and Physical Sciences Research Council (grant number EP/M506679/1) and the University of Warwick for the financial support of my studies.

Many thanks to Kengo Kamatani and Anthony Lee for taking their time to thoroughly read and examine my work, and also for making useful suggestions to improve the thesis.

Declarations

I hereby declare that the present thesis has been written by myself and that the work has not been submitted for any other degree or professional qualification. All the content was obtained by legal means. Every effort has been made to indicate and reference clearly where the work of others has been used.

Abstract

In the present thesis, we close a methodological gap of optimising the basic Markov Chain Monte Carlo algorithms. Similarly to the straightforward and computationally efficient optimisation criteria for the Metropolis algorithm acceptance rate (and, equivalently, proposal scale), we develop criteria for optimising the selection probabilities of the Random Scan Gibbs Sampler. We develop a general purpose Adaptive Random Scan Gibbs Sampler, that adapts the selection probabilities, gradually, as further information is accrued by the sampler. We argue that Adaptive Random Scan Gibbs Samplers can be routinely implemented and substantial computational gains will be observed across many typical Gibbs sampling problems.

Additionally, motivated to develop theory to analyse convergence properties of the Adaptive Gibbs Sampler, we introduce a class of Adapted Increasingly Rarely Markov Chain Monte Carlo (AirMCMC) algorithms, where the underlying Markov kernel is allowed to be changed based on the whole available chain output, but only at specific time points separated by an increasing number of iterations. The main motivation is the ease of analysis of such algorithms. Under regularity assumptions, we prove the Mean Square Error convergence, Weak and Strong Laws of Large Numbers, and the Central Limit Theorem and discuss how our approach extends the existing results. We argue that many of the known Adaptive MCMC algorithms may be transformed into the corresponding Air versions and provide an empirical evidence that performance of the Air version remains virtually the same.

List of Algorithms

1	Metropolis-Hastings	3
2	Gibbs sampler	6
3	Metropolis-within-Gibbs	7
4	AirMCMC Sampler	15
5	Adaptive Random Scan Gibbs Sampler (general idea)	27
6	Subgradient optimisation algorithm	29
7	Adaptive Gibbs Sampler based on subgradient optimisation method (not implementable)	30
8	Projection on Δ_s^ε	31
9	Adaptive Random Scan Gibbs Sampler (final version)	35
10	Random Walk Metropolis-within-Gibbs	36
11	Adaptive Random Walk Metropolis-within-Gibbs	37
12	Adaptive Random Walk Metropolis within Adaptive Gibbs	37
13	Modified AMCMC	41
14	Adaptive Random Scan Gibbs Sampler (ergodic modification)	44
15	Parallel versions of ARSGS and ARWMwAG	58
16	AirRWM	72
17	Modified AirMCMC Sampler	82
18	Randomised AirMCMC Sampler	87
19	AMCMC with the SLLN but failing convergence in distribution	88

Abbreviations

- AirMCMC – Adapted Increasingly Rarely Markov Chain Monte Carlo
- ACF – Autocorrelation Function
- AMCMC – Adaptive Markov Chain Monte Carlo
- ARSGS – Adaptive Random Scan Gibbs Sampler
- ARWM – Adaptive Random Walk Metropolis
- ARWMwAG – Adaptive Random Walk Metropolis within Adaptive Gibbs
- ARWMwG – Adaptive Random Walk Metropolis within Gibbs
- CLT – Central Limit Theorem
- DUGS – Deterministic Update Gibbs Sampler
- HMC – Hamiltonian Monte Carlo
- KAMH – Kernel Adaptive Metropolis-Hastings
- LLN – Law of Large Numbers
- MALA – Metropolis Adjusted Langevin Algorithm
- MCMC – Markov Chain Monte Carlo
- MSE – Mean Squared Error
- MwG – Metropolis within Gibbs

- PHM – Poisson Hierarchical Model
- RSGS – Random Scan Gibbs Sampler
- RWM – Random Walk Metropolis
- RWMwG – Random Walk Metropolis within Gibbs
- SLLN – Strong Law of Large Numbers
- TMVN – Truncated Multivariate Normal
- WLLN – Weak Law of Large Numbers

Chapter 1

Introduction

1.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods are a class of tools to sample from a generic probability distribution which are widely used in virtually any field where one needs to deal with uncertainty (see e.g., [Gelman et al. \[2004\]](#), [Liu \[2001\]](#), [Robert and Casella \[2004\]](#) for various examples). These methods are of particular interest in Bayesian statistical inference, where one has to estimate a model parameter as an integral with respect to a posterior distribution. More generally, in many scientific problems we are interested in computing

$$\pi(f) = \int f(x)\pi(\mathrm{d}x)$$

for various measurable functions f (see, e.g., Chapter 1 of [Liu \[2001\]](#)).

An elegant Monte Carlo method to estimate integrals $\pi(f)$ was introduced by [Metropolis and Ulam \[1949\]](#). The method is based on the idea that by generating a number of independent samples $\{X_i\}_{i=0}^{N-1}$ from the distribution π , the integral $\pi(f)$ can be estimated as

$$\hat{\pi}_N(f) := \frac{1}{N} \sum_{i=0}^{N-1} f(X_i). \quad (1.1)$$

By the Law of Large Numbers, $\hat{\pi}_N(f)$ converges to $\pi(f)$ almost surely, if the limiting integral exists. However, the distribution of interest often has a complicated structure so that the direct sampling from π is not feasible.

[Metropolis et al. \[1953\]](#) have developed a modified Monte Carlo method aimed to overcome the issue. The method is an example of a large class of Markov Chain

Monte Carlo methods. An MCMC algorithm runs a Markov chain that converges to π . Of course, we expect the Markov chain to be implementable on a computer. The output of the chain $\{X_i\}_{i=0}^N$ can then be used in the same manner as the Monte Carlo samples, i.e., we could compute (1.1) in order to estimate $\pi(f)$.

Due to the lack of computational power and limited availability of computers at the time, it took another few decades before publication of the seminal paper by [Hastings \[1970\]](#), who generalised the ideas of [Metropolis et al. \[1953\]](#) in an algorithm now known as the Metropolis-Hastings algorithm. A further push towards popularity of the MCMC methods has been done by [Geman and Geman \[1984\]](#), who developed the Gibbs Sampler which is the basic algorithm to deal with Hierarchical models (see, e.g., [Gelman et al. \[2004\]](#)). The interested reader can find a detailed historical background in [Robert and Casella \[2011\]](#).

There is a wide variety of MCMC algorithms available for users' needs. The present thesis focuses on two of the aforementioned frameworks, namely, the Metropolis-Hastings and Gibbs Sampler. As we shall show below, many of currently popular algorithms fall into the Metropolis-Hastings framework, such as Random Walk Metropolis (RWM), Metropolis-adjusted Langevin Algorithm (MALA), or Hamiltonian Monte Carlo (HMC). All of these algorithms have parameters that need to be chosen by the user. Moreover, any of them can be interlaced with the Gibbs Sampler into the Metropolis-within-Gibbs Sampler, which altogether provides users with plenty of algorithms to choose from. Below we describe the Metropolis-Hastings and Gibbs Sampler frameworks in greater detail.

1.1.1 Metropolis-Hastings framework

We assume that the reader is familiar with basic Markov Chain theory on general state spaces and refer to [Meyn and Tweedie \[2009\]](#) for the main definitions and results. The basic concepts of the MCMC theory can be found in [Roberts and Rosenthal \[2004\]](#). Having this remark in mind, we outline the Metropolis-Hastings framework below.

Let π be a probability distribution of interest on a state space \mathcal{X} with countably generated σ -algebra. Often, \mathcal{X} is a subset of a d -dimensional Euclidean space \mathbb{R}^d . Let $Q(x, \cdot)$ be essentially any Markov kernel on \mathcal{X} . For practical reasons, Q should be chosen so that for any $x \in \mathcal{X}$, it is possible sample from $Q(x, \cdot)$.

We assume that both π and $Q(x, \cdot)$ have densities with respect to (w.r.t.) some reference measure φ on \mathcal{X} (usually, φ is the Lebesgue measure on \mathbb{R}^d). Without causing ambiguity, let $\pi(x)$ and $q(x, y)$ denote the corresponding densities.

Algorithm 1 is the Metropolis-Hastings sampler that proceeds by generating

a Markov chain using the kernel Q with an additional *acceptance-rejection* procedure at each iteration, that ensures π is a stationary distribution of the underlying Markov chain.

Algorithm 1: Metropolis-Hastings

Set some initial values for $X_0 \in \mathcal{X}$; $n := 0$.

Beginning of the loop

1. Sample a proposal $Y \sim Q(X_n, \cdot)$;
2. Compute acceptance ratio $\alpha = \alpha(X_n, Y) = \min \left\{ 1, \frac{\pi(Y)q(Y, X_n)}{\pi(X_n)q(X_n, Y)} \right\}$;
3. With probability α accept the proposal and set $X_{n+1} = Y$, otherwise, reject the proposal and set $X_{n+1} = X_n$;
4. $n := n + 1$;

Go to **Beginning of the loop**

The target distribution π is only utilised when computing the acceptance ratio α in Step 2. Since α involves a ratio of π at two points, the user needs to know the density π only up to a normalising constant. The acceptance ratio α is specifically constructed to ensure that the algorithm produces a reversible chain w.r.t. π and thus, implying that π is a stationary distribution of the chain (see Propositions 1 and 2 of [Roberts and Rosenthal \[2004\]](#)).

It is possible to use a different version of the acceptance ratio in Step 2 (see an algorithm by [Barker \[1965\]](#) and a discussion by [Tierney \[1998\]](#)) but for the purpose of present thesis we restrict ourselves to the Metropolis-Hastings framework of Algorithm 1. Many of the well-known popular algorithms fall into this framework:

- *Random Walk Metropolis (RWM)*. Here the kernel density has a property $q(x, y) = q(y, x)$. Often $Q(x, \cdot) \sim N(x, \Sigma)$ is chosen to be a normal distribution centred at x with some user-defined covariance structure Σ .
- *Metropolis-adjusted Langevin Algorithm (MALA)*. Here we assume that the support of the target distribution π is the Euclidean space $\mathcal{X} = \mathbb{R}^d$ and that $\log \pi$ exists and differentiable. The proposal is generated using a single step of the discretised Langevin dynamics (see, [Roberts and Tweedie \[1996\]](#)):

$$Y \sim N \left(X_n + \sigma^2/2 \nabla \log \pi(X_n), \sigma^2 I_d \right), \quad (1.2)$$

where $\sigma > 0$ is a user-defined parameter, ∇ is the gradient operator, and I_d is

the identity matrix of dimension d .

- *Hamiltonian Monte Carlo (HMC)*. The algorithm is based on the discretisation of Hamiltonian dynamics for a target distribution π supported on \mathbb{R}^d (see [Duane et al. \[1987\]](#), [Neal \[2011\]](#), [Hoffman and Gelman \[2014\]](#)). We assume that $\log \pi$ exists and differentiable. The Markov chain (X_n, r_n) corresponding to the algorithm evolves on the extended space \mathbb{R}^{2d} and has a stationary distribution $\pi \times N(0, I_d)$, i.e., a product distribution of π and the standard multivariate normal distribution. The proposal (Y, r) is generated in three steps:

1. Generate $r \sim N(0, I_d)$, set $Y = X_n$;
2. For user-defined constants $L \in \mathbb{N}$, $\varepsilon > 0$, evolve (Y, r) L times using the leapfrog integrator:

$$\begin{aligned} r &:= r + \frac{\varepsilon}{2} \nabla \log \pi(Y); \\ Y &:= Y + \varepsilon r; \\ r &:= r + \frac{\varepsilon}{2} \nabla \log \pi(Y); \end{aligned}$$

3. Set $r := -r$.

It turns out that the corresponding density of the proposal is symmetric, i.e., $q((x, r), (\tilde{x}, \tilde{r})) = q((\tilde{x}, \tilde{r}), (x, r))$, so that the acceptance ratio in Step 2 of Algorithm 1 simplifies to $\frac{\pi(Y) \exp(-r^2/2)}{\pi(X_n) \exp(-r_n^2/2)}$.

Remark. The presented MALA and HMC samplers are not in their most generic form. A non-isotropic and non-constant diffusion matrix can be used for MALA [Roberts and Stramer \[2002\]](#), [Stramer and Tweedie \[1999\]](#), i.e., the proposal covariance matrix can take the form $M = M(X_n)$ in (1.2); a non-isotropic proposal $N(0, M)$ can be used in Step 1 of the HMC [Neal \[2011\]](#). For further discussions we refer the reader to [Girolami and Calderhead \[2011\]](#).

From practical point of view, it is important to have a guarantee of convergence for the MCMC algorithms. The basic type of convergence for Markov chains is the *convergence in distribution* or *total variation convergence*.

Definition. We say that the MCMC is ergodic if it converges in distribution, i.e.,

$$\lim_{n \rightarrow \infty} \|\mathcal{L}(X_n | X_0 = x) - \pi\|_{\text{TV}} = 0 \quad \pi - a.s., \quad (1.3)$$

where $\mathcal{L}(X_n|X_0 = x)$ is the distribution law of the chain at iteration n started at location x , and for a signed measure ν on \mathcal{X} , $\|\nu\|_{\text{TV}} := \sup_A |\nu(A)|$, where the supremum is taken over all measurable sets.

Since π is the stationary distribution of the Metropolis-Hastings Algorithm 1, ergodicity follows once we show that the chain visits all measurable sets infinitely often (φ -irreducibility) and does not demonstrate periodic behaviour (for precise statements, see Theorem 13.0.1 of [Meyn and Tweedie \[2009\]](#) or Theorem 4 of [Roberts and Rosenthal \[2004\]](#)).

It turns out that the RWM is ergodic under very weak assumptions on the proposal density q for essentially any target distribution, which follows from [Roberts and Smith \[1994\]](#) (for example, when $Q(x, \cdot)$ is the normal proposal $N(x, \Sigma)$ and π has a Lebesgue density). If $\log \pi$ is differentiable, then MALA is ergodic, which follows from [Roberts and Tweedie \[1996\]](#). HMC, however, may exhibit a periodic behaviour and thus, fail to converge without additional assumptions (see ergodicity section of [Neal \[2011\]](#)). Discussion on the sufficient conditions that imply ergodicity of the HMC are presented in a recent paper by [Durmus et al. \[2017\]](#).

1.1.2 Gibbs Sampler

Assume that the state space admits a partition $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_d$ into d components (e.g., $\mathcal{X} = \mathbb{R}^d$). For each $i \in \{1, \dots, d\}$ and $x = (x_1, \dots, x_d) \in \mathcal{X}$, let $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$.

For many Bayesian hierarchical models, it is typically infeasible to sample from the posterior directly. On the other hand, the state space has a natural partition $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_d$, in which full conditional distributions $\pi(x_i|x_{-i})$ belong to known families of distributions which one can sample from. A number of examples can be found in Chapter 15 of [Gelman et al. \[2004\]](#) or Chapter 10 of [Robert and Casella \[2004\]](#).

As was noticed by [Geman and Geman \[1984\]](#), it is possible to construct an ergodic (under mild assumptions) Markov Chain that alternates between using different full conditionals $\pi(x_i|x_{-i})$ to update the state of the chain. The corresponding MCMC algorithm is called *Gibbs Sampler*. It has been popularised in statistical community by [Gelfand and Smith \[1990\]](#), where the authors have discussed the applicability of the algorithm in Bayesian hierarchical modelling.

There are two different formal approaches to the Gibbs Sampler. *Deterministic Update Gibbs Sampler (DUGS)* updates each component x_i in a sequential order. Alternatively, *Random Scan Gibbs Sampler (RSGS)* chooses a coordinate i at ran-

dom at each iteration according to a user-supplied probability vector $p = (p_1, \dots, p_d)$. Algorithm 2 is a unified Gibbs Sampling framework.

Algorithm 2: Gibbs sampler

Set some initial values for $X_0 \in \mathcal{X}_1 \times \dots \times \mathcal{X}_d$, $n := 0$. Let

$s : \mathbb{N}_0 \rightarrow \{1, \dots, d\}$ be a coordinate selection map, which is either

- DUGS: $s(n) = 1 + (n \bmod d)$;
- RSGS: $s(n)$ randomly selects $i \in \{1, \dots, d\}$ according to a user-defined probability vector $p = (p_1, \dots, p_d)$.

Beginning of the loop

1. Compute $i = s(n)$;
2. $Y \sim \pi(X_{n,i} | X_{n,-i})$;
3. Set $X_{n+1} = (X_{n,1}, \dots, X_{n,i-1}, Y, X_{n,i+1}, \dots, X_{n,d})$;
4. $n := n + 1$;

Go to **Beginning of the loop**

Sufficient conditions for the ergodicity of the Gibbs Sampler are presented in [Roberts and Smith \[1994\]](#). Despite being based on the same idea, DUGS and RSGS may exhibit significantly different performance properties, as studied in [Roberts and Sahu \[1997b\]](#). Convergence rate in (1.3) of the DUGS depends on the order in which one updates through the full conditionals $\pi(x_i | x_{-i})$, while, as we discuss in Section 2.1 of Chapter 2, the rate of convergence of the RSGS depends on the selection probabilities (p_1, \dots, p_d) .

The careful reader can notice that the Gibbs Sampler can be put into Metropolis-Hastings framework. Indeed, let $Pr_i(x, \cdot)$ be a Markov kernel that updates x_i using its full conditional distribution $\pi(x_i | x_{-i})$. Then Step 2 of the Gibbs Sampler may be seen as the proposal generating Step 1 of the Metropolis algorithm 1. One can easily see that the acceptance ratio α in this case is equal to 1, meaning that the proposal should always be accepted in Step 3 of the Gibbs Sampler.

From the author's point of view, it is better to think of Gibbs Sampler as a meta-sampler. More precisely, with an additional accept-reject step, it is possible to utilise any proposal $Q_{x_{-i}}(x_{-i}, \cdot)$ in Step 2 of the Gibbs Sampler instead of the full conditional proposal. The resulting Algorithm 3 is called Metropolis-within-Gibbs (MwG), or Metropolised Gibbs Sampler as in [Robert and Casella \[2004\]](#). The MwG allows the user to explore the different dimensions of the state space using different

proposal distributions. As we shall see in Section 2.7.3 of Chapter 2 it might be advantageous to use MwG even when one can sample from the full conditionals.

Algorithm 3: Metropolis-within-Gibbs

Set some initial values for $X_0 \in \mathcal{X}_1 \times \dots \times \mathcal{X}_d$, $n := 0$. Let

$s : \mathbb{N}_0 \rightarrow \{1, \dots, d\}$ be a coordinate selection map, which is either

- DUGS: $s(n) = 1 + (n \bmod d)$;
- RSGS: $s(n)$ randomly selects $i \in \{1, \dots, d\}$ according to a user-defined probability vector $p = (p_1, \dots, p_d)$.

Beginning of the loop

1. Compute $i = s(n)$;
2. $Y \sim Q_{X_{n,-i}}(X_{n,-i}, \cdot)$;
3. Compute acceptance ratio $\alpha = \min \left\{ 1, \frac{\pi(Y|X_{n,-i})q_{X_{n,-i}}(Y, X_n)}{\pi(X_{n,i}|X_{n,-i})q_{X_{n,-i}}(X_n, Y)} \right\}$;
4. With probability α accept the proposal and set
$$X_{n+1} = (X_{n,1}, \dots, X_{n,i-1}, Y, X_{n,i+1}, \dots, X_{n,d}),$$
otherwise, reject the proposal and set $X_{n+1} = X_n$;
5. $n := n + 1$;

Go to **Beginning of the loop**

1.2 Adaptive MCMC

The idea behind the MCMC technique is fairly simple. First, choose a Markov kernel P with stationary distribution π . Then run a Markov chain with the kernel P and use the chain output $\{X_i\}_{i=1}^n$ to estimate integrals $\int f(x)d\pi(dx)$ by the average $\hat{\pi}_N$ in (1.1). While we have already seen that designing valid kernels P is easy, it is a hard problem to identify the ones for which $\hat{\pi}_N$ does not converge excessively slow. Typically, one would re-run the MCMC algorithm many times before an optimal Markov kernel P is found.

It is, of course, impossible to identify the best kernel P that minimises the running time of the corresponding MCMC algorithm. In practice, one rather encounters a problem of choosing the best Markov kernel out of a parametrised subclass P_γ , $\gamma \in \Gamma$, should it be the best proposal covariance Σ for the RWM or the scaling

parameter σ for MALA. Usually the optimal parameter γ is unknown *a priori*, as it depends on the intractable distribution π in a complicated way.

A more attractive alternative to hand tuning is to design an automated algorithmic procedure that would adjust γ indefinitely, as further information accrues from the chain output. Formally, this approach is called *adaptive MCMC* (AMCMC) (see, e.g., [Andrieu and Thoms \[2008\]](#), [Roberts and Rosenthal \[2007\]](#), [Rosenthal \[2011\]](#)). AMCMC produces a chain X_n by repeating the following two steps:

- (1) Sample X_{n+1} from $P_{\gamma_n}(X_n, \cdot)$;
- (2) Given $\{X_0, \dots, X_{n+1}, \gamma_0, \dots, \gamma_n\}$, update γ_{n+1} according to some adaptation rule.

After running an adaptive chain, we can use its output in the same way as if it were a usual MCMC output (e.g., compute $\hat{\pi}_N$ to estimate $\int f d\pi$).

Ultimately, we encounter two equally important issues. First, we need to construct the adaptation rule in Step (2). Secondly, the output chain $\{X_n\}$ of AMCMC algorithms is usually not Markov, meaning that specialised techniques should be used for their theoretical analysis.

In many basic settings, there is a constructive methodological guidance of how to hand tune γ based on a pilot MCMC run.

- *Optimal scaling of the RWM algorithm.* The state space is assumed to be Euclidean $\mathcal{X} = \mathbb{R}^d$. The optimal covariance matrix Σ of the proposal $Q(x, \cdot) \sim N(x, \Sigma)$ is $\Sigma = \sigma^2 \Sigma_\pi$, where Σ_π is the covariance matrix of π and σ^2 is such that the average acceptance ratio

$$\alpha_{\text{ave}} := \int \alpha(x, y) Q(x, dy) \pi(dx) \quad (1.4)$$

is equal to 0.44 if the state space is one dimensional (i.e., $d = 1$) and 0.234 if $d \geq 5$.

The above optimal scaling in one dimensional case follows from the experimental results of [Gelman et al. \[1996\]](#) for the standard normal target distribution. For large values of d ($d \geq 5$ as suggested in [Rosenthal \[2011\]](#)) and a proposal of the form $Q(x, \cdot) \sim N(x, \sigma^2 I_d)$, [Roberts et al. \[1997\]](#) prove that under strong assumption on the target distribution (it should be a product of d i.i.d. random variables), stochastic process $U_t = X_{\lfloor td \rfloor, 1}$ (the first coordinate of $X_{\lfloor td \rfloor}$) behaves like a diffusion (see Theorem 1.2 of [Roberts et al. \[1997\]](#)). The speed of convergence of the diffusion to the stationary distribution is maximised for

σ^2 , that results in the average acceptance ratio (1.4) being 0.234. Moreover, for large values of d , the optimal value of σ^2 is $\frac{2.38^2}{d}$. The assumption of i.i.d. structure of the target distribution was relaxed by Roberts and Rosenthal [2001], Bédard [2007] and generalised to the multivariate normal target with a covariance matrix Σ_π and proposal $Q(x, \cdot) \sim N(x, \Sigma)$ (see also Roberts and Rosenthal [2009], Rosenthal [2011]).

While the results seem to be too restrictive, the suggested scaling of the normal proposal covariance to retain the average acceptance ratio (1.4) around 0.234 seems to be very robust and useful in many applications. Roberts and Rosenthal [2001] demonstrate that the proposed scaling of the RWM is optimal in certain settings even on discrete spaces. Bédard and Rosenthal [2008] provide in-depth discussion of the optimality result, while useful practical advice can be found in Section 4.2.6 of Rosenthal [2011].

- *Optimal scaling of MALA.* In this case for $d \geq 5$ the optimal scaling σ^2 of the proposal $Q(x, \cdot) \sim N(x + \sigma^2/2\nabla \log \pi(x), \sigma^2 I_d)$ is such that the average acceptance ratio (1.4) is 0.574.

In the same manner as for the RWM, Roberts and Rosenthal [1998] have shown that under certain strong conditions (π is a product of d i.i.d. random variables), a process $U_t := X_{\lfloor nt/(d^{1/3}) \rfloor, 1}$ has a diffusion limit and the speed of convergence of the diffusion is maximised precisely when the average acceptance ratio (1.4) is 0.574. Non-i.i.d. target π has been considered by Breyer et al. [2004] for “mean field” models and in the most general case by Pillai et al. [2012], where the authors have proved that under certain conditions on the covariance structure of π , U_t converges to a diffusion on a Hilbert space with the same optimal average acceptance ratio of 0.574.

- *Optimal HMC parameters for a fixed integration time.* The author is not aware of any results for optimal choice of both the discretisation parameter ε and the number of Leapfrog steps L simultaneously.

On the other hand, in a high dimensional Euclidean space \mathbb{R}^d , for a fixed integration time T (i.e., for a fixed integration time T and given discretisation ε , the number of Leapfrog steps is $L := \lceil T/\varepsilon \rceil$), the optimal value of ε is such that results in the average acceptance ratio (1.4) being equal to 0.651.

The result is established by Beskos et al. [2013] in the i.i.d. scenario (i.e., the target distribution π is a product of d i.i.d. random vectors), where the optimal ε is chosen to maximise the mean-squared jumping distance, which is

the same ε that retains the average acceptance ratio at 0.651.

The above optimality result, however, does not tell how to choose the integration time T . A successful attempt to overcome this issue is proposed by [Hoffman and Gelman \[2014\]](#), where the authors construct an HMC based sampler called *No-U-Turn Sampler*. We shall not further discuss the algorithm, since it does not fall into the Metropolis-Hastings framework, and refer the reader to the original paper for details.

In all of the proposed algorithms, the optimal scaling parameter is expressed in terms of the average acceptance ratio α_{ave} (1.4), which depends on the target distribution π . Therefore, in order to find the optimal scaling parameter, at every iteration of an optimisation algorithm we need to run an MCMC algorithm that estimates α_{ave} , which is very inefficient in practice. On the other hand, the optimisation algorithm can be very naturally incorporated into Step (2) of the Adaptive MCMC framework. It is proposed by [Roberts and Rosenthal \[2009\]](#), [Andrieu and Thoms \[2008\]](#) to use Robbins-Monro stochastic optimiser [Robbins and Monro \[1951\]](#) to learn optimal scaling σ on the fly by iteratively updating

$$\sigma_{n+1} := \sigma_n \cdot \exp(r_n(\alpha_{\text{optimal}} - \alpha)), \quad (1.5)$$

where α_{optimal} is the target acceptance ratio (e.g., 0.234 for RWM in high dimensions), α is the acceptance ratio 2 at n -th iteration of Algorithm 1; $r_n > 0$ is the *learning rate*, such that $\lim_{n \rightarrow \infty} r_n = 0$. Condition $\sum_{n=1}^{\infty} r_n = \infty$ ensures that the adaptations are done infinitely often, preventing σ_n from converging to a wrong value (see Section 5.1.2 of [Andrieu and Thoms \[2008\]](#)).

Based on the optimal scaling results described above and the scaling procedure (1.5), a variety of AMCMC algorithms has been developed, including, among others, the Adaptive Metropolis (AM) [Haario et al. \[2001\]](#), [Roberts and Rosenthal \[2009\]](#), [Vihola \[2012\]](#), Adaptive MALA [Atchadé \[2006\]](#), [Marshall and Roberts \[2012\]](#), Adaptive Metropolis-within-Gibbs [Roberts and Rosenthal \[2009\]](#), or samplers specialised to model selection [Nott and Kohn \[2005\]](#), [Griffin et al. \[2017\]](#).

Empirically, Adaptive MCMC methods largely outperform their non-adaptive counterparts, often by a factor exponential in dimension, and enjoy great success in many challenging applications (see e.g. [Solonen et al. \[2012\]](#), [Bottolo and Richardson \[2010\]](#)). Nevertheless, despite the large body of work that we discuss in Section 3.3 of Chapter 3, the theoretical underpinning of AMCMC is lagging behind that of non-adaptive MCMC. The AMCMC algorithms are notoriously difficult to analyse

due to their intrinsic non-Markovian dynamics resulting from alternating Steps (1) and (2) above.

The first guidance on how to construct an ergodic AMCMC is proposed by Gilks et al. [1998], where the authors allow any kind of adaptations to take place but only at regeneration times (3.12) of the underlying Markov chains. Unfortunately, the algorithm is inefficient in high dimensional settings since the regeneration rate deteriorates exponentially in dimension. We shall recycle ideas of Gilks et al. [1998] in Chapter 3, where we develop a new methodology for AMCMC.

More practical conditions, introduced by Roberts and Rosenthal [2007], are known as *diminishing* and *containment* conditions:

(C1) *Diminishing adaptation condition.*

$$\sup_{x \in \mathcal{X}} \|P_{\gamma_n}(x, \cdot) - P_{\gamma_{n+1}}(x, \cdot)\|_{TV} \xrightarrow{P} 0 \text{ as } n \rightarrow \infty,$$

where $\|\cdot\|_{TV}$ is the total variation norm, $\gamma_n \in \Gamma$ - random sequence of parameters, and \xrightarrow{P} denotes the convergence in probability.

(C2) *Containment condition.* For $x \in \mathbb{R}^d$, $\gamma \in \Gamma$ and all $\varepsilon > 0$ define a function

$$M_\varepsilon(x, \gamma) := \inf \left\{ N \geq 1 \left| \|P_\gamma^N(x, \cdot) - \pi(\cdot)\|_{TV} \leq \varepsilon \right. \right\}.$$

We say that an adaptive chain $\{X_n, \gamma_n\}$ started at (X_0, γ_0) satisfies containment condition if for all $\varepsilon > 0$, the sequence $\{M_\varepsilon(X_n, \gamma_n)\}_{n=0}^\infty$ is bounded in probability, i.e., $\lim_{N \rightarrow \infty} \sup_n \mathbb{P} \left(M_\varepsilon(X_n, \gamma_n) > N \right) = 0$, where \mathbb{P} is the probability measure induced by the chain started at (X_0, γ_0) .

Theorem 2 of Roberts and Rosenthal [2007]. *Let $\{X_n, \gamma_n\}$ be an adaptive chain with $\{\gamma_n\}$ being the corresponding sequence of parameters. If $\{X_n, \gamma_n\}$ satisfies (C1) and (C2), then the adaptive chain is ergodic, i.e.,*

$$\|\mathcal{L}(X_n|X_0, \gamma_0) - \pi\|_{TV} \rightarrow 0 \text{ as } n \rightarrow \infty,$$

where $\mathcal{L}(X_n)$ is the probability distribution law of X_n and π is the target distribution.

Condition (C1) can often be easily verified for many practical problems (e.g., when estimating the optimal scaling parameter through (1.5)). The condition only requires adaptations to happen at a decreasing rate. Where it does not hold, it is

easy to enforce the condition. For example, at every iteration n of the adaptive chain, we can flip a coin with heads probability p_n and adapt the parameter γ_n in Step (2) only if the coin heads. If $\lim_{n \rightarrow \infty} p_n = 0$, then (C1) holds. Note that the sequence p_n may decay arbitrarily slowly.

Condition (C2), however, is very technical and hard to verify in practice. The condition controls uniform convergence over the parameter space to the stationary distribution. It is shown by [Latuszyński and Rosenthal \[2014\]](#), that if an adaptive algorithm satisfies (C1) but fails the containment condition, it performs worse than any non-adaptive algorithm for any kernel P_γ , $\gamma \in \Gamma$. Therefore, the containment condition is in some sense intrinsic to a successful AMCMC algorithm.

There has been a lot of effort put into developing practical conditions that guarantee the containment (C2). The most up-to-date results are due to [Bai et al. \[2011\]](#). The key assumptions are *simultaneous geometric (polynomial) drift* Assumptions 2 and 3, which are presented in Section 3.2 of Chapter 3, where we also introduce a novel *local simultaneous geometric drift* Assumptions 4 to deal with a wider class of adaptive MCMC algorithms (such as the Adaptive Gibbs Sampler). The major part of the research concerning ergodicity of AMCMC algorithms deals with developing easily verifiable conditions that imply the aforementioned assumptions (see, e.g., [Haario et al. \[2001\]](#), [Latuszyński et al. \[2013b\]](#), [Atchadé and Rosenthal \[2005\]](#), [Andrieu and Moulines \[2006\]](#), [Saksman and Vihola \[2010\]](#)).

For Markov chains, ergodicity and the Strong Law of Large Numbers (SLLN) for the averages (1.1) are verified under the same assumptions (see Theorems 13.0.1 and 17.0.1 (i) of [Meyn and Tweedie \[2009\]](#)). On the contrary, for AMCMC, the containment and diminishing adaptation conditions do not guarantee the SLLN, as shown in a counter Example 4 of [Roberts and Rosenthal \[2007\]](#). Various additional sufficient conditions for the SLLN were considered by [Atchadé and Fort \[2010\]](#), [Vihola \[2012, 2011\]](#), [Saksman and Vihola \[2010\]](#), [Atchadé et al. \[2011\]](#).

It is typical to use drift conditions in the proof of the Central Limit Theorem (CLT) for Markov chains (see, e.g., Theorem 17.0.1 (iii) of [Meyn and Tweedie \[2009\]](#) or the results of [Latuszyński et al. \[2013a\]](#)), making Assumptions 2, 3 and 4 from Chapter 3 look natural. Nevertheless, even under the simultaneous drift conditions, proving the CLT for adaptive MCMC seems particularly hard. [Andrieu and Moulines \[2006\]](#) introduce additional technical conditions to prove the CLT, where, in particular, convergence of the adapted parameters is required.

In Chapter 3 we shall develop a methodology to modify the existing AMCMC that is aimed to avoid any additional assumptions on top of the standard simultaneous drift conditions, and still guarantee the SLLN, Mean Square Error (MSE)

convergence, and CLT for the adaptive algorithms.

1.3 Overview of the thesis and main results

The main body of the present thesis consists of three chapters. In Sections 1.3.1 and 1.3.2 we describe the contributions of Chapters 2 and 3 in greater detail.

To date, there is no criteria for optimising the selection probabilities of the Random Scan Gibbs Sampler (see Algorithm 2 in Section 1.1.2 above). In Chapter 2 we close this methodological gap and develop a general purpose *Adaptive Random Scan Gibbs Sampler* that adapts the selection probabilities.

We present a number of moderately- and high-dimensional examples, including truncated Gaussians, Bayesian Hierarchical Models and Hidden Markov Models, where significant computational gains are empirically observed for both the Adaptive Gibbs and *Adaptive Metropolis within Adaptive Gibbs* version of the algorithm. We argue that the Adaptive Random Scan Gibbs Sampler can be routinely implemented and substantial computational gains will be observed across many typical Gibbs sampling problems. We introduce the *local simultaneous polynomial drift* condition (2.34) that relaxes the commonly used simultaneous geometric drift condition (2.33) and ensures ergodicity of a larger class of modified AMCMC presented in Algorithm 13, and, in particular, of the Adaptive Gibbs Sampler Algorithm 14.

In Chapter 3 we develop a class of *Adapted Increasingly Rarely Markov Chain Monte Carlo* (AirMCMC) algorithms where the underlying Markov kernel is allowed to be changed based on the whole available chain output, but only at specific time points, separated by an increasing number of iterations. The main motivation is the ease of analysis of such algorithms. Under assumption of either simultaneous (3.9) or (weaker) local simultaneous (3.11) geometric drift condition, or simultaneous polynomial drift (3.10), we prove the Strong and Weak Laws of Large Numbers (SLLN, WLLN), the Central Limit Theorem (CLT), quantify the rate of Mean-Square Error (MSE) decay, and discuss how our approach extends the existing results. We argue that many of the known AMCMC algorithms may be transformed into an Air version and provide an empirical evidence that the performance of the Air versions remains virtually the same.

In Chapter 4 we describe a C++ implementation of Adaptive Metropolis-within-Gibbs framework. The library is handy for writing Metropolis-within-Gibbs type algorithms and can be used to reproduce the simulations from the present thesis.

1.3.1 Adaptive Gibbs Sampler

The RSGS and MwG Algorithms 2 and 3 are very popular in practice. Recall that at every iteration the RSGS chooses a coordinate i with probability p_i and updates it from its full conditional distribution. Usually, uniform selection probabilities p_i are used, while we argue that this is often a sub-optimal strategy. To date, there is no guidance on the optimal choice of the selection probabilities (see [Latuszyński et al. \[2013b\]](#)).

A possible solution is to use those probabilities that maximise the L_2 —spectral gap (hereafter, spectral gap) (2.2) of the corresponding algorithm. Of course, estimating the spectral gap is a challenging problem. On the other hand, if the target distribution is normal, then for the RSGS, there is an explicit formula for the spectral gap (2.4). Since (2.4) depends only on the correlation structure of the target distribution, the equation (2.4) may be optimised for an arbitrary target distribution resulting in some selection probabilities p^{opt} that we call *pseudo-optimal*.

In Bayesian Analysis, by virtue of Bernstein-von Mises Theorem (see, e.g., [van der Vaart \[2000\]](#)), under certain conditions and given sufficient amount of observations, the posterior distribution is well approximated by an appropriate Gaussian. Thus, if one applies the RSGS to sample from the posterior, the pseudo-optimal weights p_i might represent a good approximation of the true optimal weights that maximise the spectral gap. Moreover, as we demonstrate by simulations in Section 2.7, even if the target distribution is far from normal or not continuous, the pseudo-optimal weights might still be advantageous over the uniform selection probabilities.

Since the pseudo-optimal selection probabilities are a function of the correlation structure of the target distribution, which is usually not known in practice, and optimising (2.4) is a hard problem (see, e.g., [Overton \[1988\]](#)), in Section 2.4 we develop a general purpose *Adaptive Random Scan Gibbs Sampler* (ARSGS) that adapts the selection probabilities on the fly.

We also find that a special case of the MwG algorithm, namely, Random Walk Metropolis within Gibbs (RWMwG) algorithm, may be significantly improved by adapting both the proposal distribution (for instance, as suggested in [Rosenthal \[2011\]](#)) and the underlying selection probabilities in the same manner as for the RSGS.

Because the implementation of the adaptive algorithms is easy and the additional computational cost is often negligible compared to the total computational effort, we argue that the algorithms could be routinely implemented. We demonstrate in Section 2.7 that the ARSGS and ARWMwAG algorithms speed up convergence to the target distribution for many typical Gibbs sampling problems.

Finally, we introduce a notion of *local simultaneous geometric drift* condition **(A3)** in Section 2.6, which is a natural property for the RSGS, as we demonstrate in Theorem 5. In Theorem 8 we prove convergence of the modified ARSGS under the local simultaneous geometric drift condition. In Section 3.2 of Chapter 3 we derive various convergence properties of (1.1) of generic AMCMC algorithms under this condition.

1.3.2 AirMCMC

In this chapter we propose to redesign adaptive MCMC algorithms so that they become more tractable mathematically, but the ability to self tune the parameters becomes unaffected.

We develop *Adaptive Increasingly Rarely MCMC (AirMCMC)* framework, where adaptations of the underlying Markov kernel P_γ are only allowed to happen at scheduled times with an increasing lag between them. Denote the consecutive lags as $n_k \nearrow \infty$ and the adaptation times as

$$N_j := \sum_{k=1}^j n_k, \quad \text{with } N_0 = n_0 := 0. \quad (1.6)$$

The generic design of an AirMCMC is presented in Algorithm 4.

Algorithm 4: AirMCMC Sampler

Set some initial values for $X_0 \in \mathcal{X}$; $\gamma_0 \in \Gamma$; $\bar{\gamma} := \gamma_0$; $k := 1$; $n := 0$.

Beginning of the loop

1. For $i = 1, \dots, n_k$
 - 1.1. sample $X_{n+i} \sim P_{\bar{\gamma}}(X_{n+i-1}, \cdot)$;
 - 1.2. given $\{X_0, \dots, X_{n+i}, \gamma_0, \dots, \gamma_{n+i-1}\}$ update γ_{n+i} according to some adaptation rule.
2. Set $n := n + n_k$, $k := k + 1$. $\bar{\gamma} := \gamma_n$.

Go to **Beginning of the loop**

Note that Step 1.2 of the AirMCMC pseudo code allows a background pre-computation of the parameter γ , analogous to that in Step (2) of AMCMC. However, the dynamics of $\{X_n\}$ is driven by $P_{\bar{\gamma}}$, and the value of $\bar{\gamma}$ is updated at the scheduled times N_j only. It is intuitively clear that updating the transition kernel at every step is not necessary for efficient tuning because the new information about

the optimal γ acquired from π in a single move of X_n is infinitesimal as the total length of simulation increases. We demonstrate this empirically in Section 3.1 by comparing performance of adaptive scaling and Adaptive Metropolis algorithms to their Air versions for various choices of the lag sequence $\{n_k\}$.

Theoretical analysis of AirMCMC benefits from the fact that the law

$$\mathcal{L}(X_{N_j+1}, \dots, X_{N_j+n_{j+1}} | \mathcal{G}_j), \quad \text{where } \mathcal{G}_j := \sigma(X_0, \dots, X_{N_j}, \gamma_0, \dots, \gamma_{N_j}),$$

is that of a Markov chain with transition kernel $P_{\gamma_{N_j}}$. Consequently, the standard Markov chain arguments apply to individual epochs between adaptations of increasing length n_k . In Section 3.2 we state that AirMCMC algorithms preserve the main convergence properties, namely, the WLLN, SLLN and the CLT. Also, we show that MSE of $\hat{\pi}_N(f)$ decays to 0 at a rate that is arbitrary close or equal to $1/N$ and with constants that in principle can be made explicit. We establish these results under regularity conditions that are standard for MCMC and AMCMC analysis, namely simultaneous geometric drift conditions of $\{P_\gamma\}_{\gamma \in \Gamma}$, (MSE, WLLN, SLLN, CLT) and simultaneous polynomial drift conditions (WLLN, SLLN, CLT), as well as assuming a weaker (and non-standard) local simultaneous geometric drift conditions (MSE, WLLN, SLLN, CLT). No further technical assumptions are needed, in particular, neither diminishing adaptation, nor Markovianity of the bivariate process (X_n, γ_n) that are typically required in theoretical analysis of the AMCMC. A detailed discussion of how these results relate to available AMCMC theory is provided in Section 3.3. Proofs of the theoretical properties of AirMCMC are gathered in Section 3.6.

There are many other advanced MCMC algorithms that are outside the scope of this thesis, in particular, the Scalable Langevin Exact Algorithm [Pollock et al. \[2016\]](#), Zig-Zag algorithm [Bierkens and Duncan \[2017\]](#), or non-reversible MALA [Ottobre et al. \[2017\]](#). While we do not discuss these algorithms, it is worth to notice that virtually any MCMC algorithm may be put into the AirMCMC framework for the adaptation purpose as long as one has the adaptation rule for the Step 1.2.

We provide a case study in Section 3.1, where we demonstrate that a careful choice of the sequence of lags $\{n_k\}$ between the adaptations does not slow down convergence of the corresponding AMCMC. In fact, Air versions of the algorithms may significantly reduce the total computational time, since less resources are spent on adaptation.

In Section 3.4 we demonstrate how AirMCMC helps establish theoretical underpinning of advanced algorithms. We consider the Adaptive Random Scan

Gibbs Sampler (ARSGS) and the recently proposed Kernel Adaptive Metropolis Hastings (KAMH) [Sejdinovic et al. \[2014\]](#) algorithms. Asymptotic properties of (1.1) for both the ARSGS and KAMH are not covered by the currently available AMCMC theory when applied to a target with unbounded support. However, for their Air versions, we establish MSE convergence, the WLLN and SLLN when applied to suitable targets. We conclude the chapter with a discussion in Section 3.5.

Chapter 2

Adaptive Gibbs Sampler

The chapter is organised as follows. In Section 2.1 we exploit ideas of Amit [1991, 1996] and Roberts and Sahu [1997a] to derive the formula for the spectral gap (2.4) for a particular case of sampling from the multivariate normal distribution using RSGS scheme. For a general target distribution we introduce the concept of *pseudo-spectral gap* and *pseudo-optimal weights* in Section 2.2 and demonstrate potential advantage of the pseudo-optimal weights over the uniform ones on toy examples studied in Section 2.3. Derivation of the Adaptive Ransom Scan Gibbs Sampler (ARS GS) and Adaptive Random Walk Metropolis within Adaptive Gibbs (ARWMwAG) algorithms is done in Sections 2.4 and 2.5, respectively. Convergence properties of the adaptive algorithms are discussed in Section 2.6. We provide simulation study and discuss computational cost of the adaptive algorithms in Section 2.7. We close the chapter with a discussion in Section 2.8. If not stated otherwise, all the proofs are given in the Section 2.9.

2.1 RSGS spectral gap for Multivariate Gaussian distribution

In this section we consider the RSGS for the normal target distribution and establish an explicit representation of the spectral gap in Theorem 1. One may skip all the technical details and notice only that the spectral gap in this case relies solely on the correlation structure of the target distribution and the selection probabilities.

Let π be a distribution of interest in \mathbb{R}^d . Let Σ and $Q = \Sigma^{-1}$ denote the covariance matrix of π and its inverse respectively, where we assume throughout the paper that Σ is positive-definite. Partition Q into blocks $Q = (Q_{ij})_{i,j=1}^s$ where Q_{ij} is a $r_i \times r_j$ matrix, $\sum_{i=1}^s r_i = d$. For vectors $x \in \mathbb{R}^d$ introduce splitting $x = (x_1, \dots, x_s)$,

where x_i is a vector in \mathbb{R}^{r_i} so that $x_i = (x_{i1}, \dots, x_{ir_i})$.

Given a probability vector $p = (p_1, \dots, p_s)$ (i.e., $p_i > 0$, $\sum_{i=1}^s p_i = 1$), $\text{RSGS}(p)$ is a Markov kernel that at every iteration chooses a subvector $x_i = (x_{i1}, \dots, x_{ir_i})$ with probability p_i and updates it from the conditional distribution $\pi(x_i | x_{-i})$ of x_i given $x_{-i} := (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_s)$. In other words, the $\text{RSGS}(p)$ is a Markov chain with kernel

$$P_p(x, A) = \sum_{i=1}^s p_i Pr_i(x, A), \quad (2.1)$$

where A is a π -measurable set, $x \in \mathbb{R}^d$ and Pr_i is a kernel that stands for updating x_i from the full conditional distribution $\pi(x_i | x_{-i})$. We call the kernel Pr since it is in fact a projection operator (i.e., $Pr^2 = Pr$) on the set of the space of square integrable functions $L_2(\mathbb{R}^d, \pi)$ with respect to π .

For π -integrable functions f , let $(P_p f)(x) := \int f(y) P_p(x, dy)$.

Definition. Let $\rho = \rho(p) > 0$ be the minimum number such that for all $f \in L_2(\mathbb{R}^d, \pi)$ and $r > \rho$,

$$\lim_{n \rightarrow \infty} r^{-2n} \mathbf{E}_\pi[\{(P_p^n f)(x) - \pi(f)\}^2] = 0. \quad (2.2)$$

Then ρ is called the L_2 -rate of convergence in $L_2(\mathbb{R}^d, \pi)$ of the Markov chain with the kernel P_p . The value $1 - \rho$ is called the L_2 -spectral gap (or simply spectral gap) of the kernel P_p .

In the case when $s = d$ and the selection probabilities are uniform, i.e., $p = (\frac{1}{d}, \dots, \frac{1}{d})$, Amit [Amit \[1996\]](#) provides a formula for the spectral gap. Here we generalise Amit's result by essentially changing p_i for $\frac{1}{s}$ in the proof of Theorem 1 in [Amit \[1996\]](#).

It is easy to see that the RSGS kernel is reversible w.r.t. the target distribution π . It is known that if the spectrum of kernel P_p (considered as an operator on $L_2(\mathbb{R}^d, \pi)$) consists of eigenvalues only, the L_2 -rate of convergence is given by the second largest eigenvalue of the kernel P_p (follows, e.g, from Theorem 2 and the following remark in [Roberts and Rosenthal \[1997\]](#)).

There are two key steps to establish an explicit formula for the rate of convergence $\rho(p)$.

Step 1. For the kernels P_p , find finite dimensional invariant subspaces S_k (i.e., $P_p S_k \subset S_k$) in $L_2(\mathbb{R}^d, \pi)$ by considering action of P_p on the orthonormal basis of Hermite polynomials.

Step 2. Identify the subspace S_k with the maximum eigenvalue less than one.

To clarify the steps we need to introduce some additional notations. Without loss of generality, suppose that π has zero mean.

Let $K = \sqrt{Q}$ be the symmetric square root of Q defined through the spectral decomposition, i.e., if for an orthogonal matrix U (i.e., $U^\top U = I_d$), $Q = U \text{diag}(\lambda_1, \dots, \lambda_d) U^\top$, then $P = U \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_d}) U^\top$. Set

$$D_i = \text{diag}(0, \dots, Q_{ii}^{-1}, \dots, 0), \quad (2.3)$$

where we stress that D_i is a $d \times d$ matrix with Q_{ii}^{-1} being at the same place as in partition $(Q_{ii}^{-1})_{i=1}^s$.

For $\alpha = (\alpha_1, \dots, \alpha_d) \in Z_+^d$ let $\alpha! = \alpha_1! \cdots \alpha_d!$, $|\alpha| = \alpha_1 + \cdots + \alpha_d$. Define h_k to be the *Hermite polynomial* of order k , i.e.,

$$h_k(x) = (-1)^k \exp\left(\frac{x^2}{2}\right) \frac{d^k}{dx^k} \exp\left(-\frac{x^2}{2}\right), \quad x \in \mathbb{R}^d.$$

Set $H_\alpha(x) = \frac{1}{\sqrt{\alpha!}} h_{\alpha_1}(x_1) \cdots h_{\alpha_d}(x_d)$, $H_0(x) := 1$. The next lemma summarises Steps 1 and 2 above.

Lemma 1. $\{H_\alpha(Kx) \mid \alpha \in Z_+^n\}$ form an orthonormal basis in $L_2(\mathbb{R}^d, \pi)$ and for all integers $k \geq 0$, spaces

$$S_k := \text{span}\left\{H_\alpha(Kx) \mid |\alpha| = k\right\},$$

spanned by $\{H_\alpha(Kx) \mid |\alpha| = k\}$, are finite dimensional and P_p -invariant (i.e., $P_p(f) \in S_k$ for all $f \in S_k$). Moreover, for all $k \geq 0$,

$$\lambda_{\max}(P_p|_{S_1}) \geq \lambda_{\max}(P_p|_{S_k}),$$

where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue and $P_p|_{S_k}$ is a restriction of P_p on S_k .

Lemma 1 immediately implies that $\text{Gap}(p) = 1 - \lambda_{\max}(P_p|_{S_1})$ and the next theorem provides a representation of $\text{Gap}(p)$ through the correlation structure of the target distribution.

Theorem 1. The L_2 -spectral gap in the RSGS(p) scheme for the Gaussian target distribution with precision matrix Q is given by

$$\text{Gap}(p) = 1 - \lambda_{\max}(F_1), \quad (2.4)$$

where

$$F_1 = I - K \left(\sum_{i=1}^s p_i D_i \right) K, \quad (2.5)$$

D_i is given by (2.3), and $K = \sqrt{Q}$.

Since S_1 is a set of linear functions, Lemma 1 also implies

Theorem 2. *Consider a Gibbs kernel P_p that corresponds to a normal target distribution π . Then the second largest eigenfunction of P_p in $L_2(\mathbb{R}^d, \pi)$ is a linear function in \mathbb{R}^d .*

We end this section by comparing formula (2.4) with the results by Roberts and Sahu [1997a]. Consider the case when $p_1 = \dots = p_s = \frac{1}{s}$ and introduce a matrix

$$A = I - \text{diag}(Q_{11}^{-1}, \dots, Q_{ss}^{-1})Q, \quad (2.6)$$

The following lemma will be useful throughout the paper and can be easily obtained.

Lemma 2. *Let A and B be two $d \times d$ matrices. Then AB and BA have the same eigenvalues.*

Lemma 2 implies that the spectrum (the set of all eigenvalues) of A defined in (2.6) is equal to the spectrum of

$$I - K \text{diag}(Q_{11}^{-1}, \dots, Q_{ss}^{-1})K.$$

One can easily see that $T^{(i)} := I - K D_i K$ is a projection matrix, hence

$$I - K \text{diag}(Q_{11}^{-1}, \dots, Q_{ss}^{-1})K = I + \sum_{i=1}^s T^{(i)} - sI \geq (1-s)I,$$

and the minimum eigenvalue of A is bounded below by $(1-s)$. Therefore, (2.4) gives

$$\text{Gap}\left(\frac{1}{s}\right) = \lambda_{\max}\left(\frac{1}{s}((s-1)I + A)\right) = \frac{1}{s}(s-1 + \lambda_{\max}(A)),$$

where $\text{Gap}\left(\frac{1}{s}\right)$ is the spectral gap of the RSGS with the uniform selection probabilities.

The last equation is the representation of the spectral gap in Theorem 2 of [Roberts and Sahu \[1997a\]](#).

2.2 Pseudo-spectral gap

For a general target distribution computing the spectral gap is not feasible. But one can always deal with its normal counterpart (2.4) which we call *pseudo-spectral gap*. Optimising (2.4) over all possible selection probabilities p leads to the notion of *pseudo-optimal selection probabilities*.

As we have discussed in Chapter 1, in many Bayesian settings Bernstein-von Mises theorem (see, e.g, Section 10.2 of [van der Vaart \[2000\]](#)) applies, that is, under certain conditions the posterior distribution converges to normal in the total variation norm. Thus, we hope that the pseudo-spectral gap of RSGS is a meaningful approximation to the true value of the spectral gap and the pseudo-optimal weights are close to the ones that maximise the spectral gap.

In fact, as we will see in Section 2.7, where we sample from the Truncated Multivariate Normal distribution and the posterior in Markov Switching Model, if the correlation matrix is well-informative about the dependency structure of the target distribution, running the RSGS with the pseudo-optimal weights instead of the uniform ones, may substantially fasten the convergence, even if the target distribution has discrete components.

To formally define the pseudo-spectral gap, we need a couple of additional notations.

$$\Delta_{s-1} := \{\bar{p} \in \mathbb{R}^{s-1} | \bar{p}_i > 0, i = 1, \dots, s-1; 1 - \bar{p}_1 - \dots - \bar{p}_{s-1} > 0\}$$

is a convex set in \mathbb{R}^{s-1} , so that Δ_{s-1} defines a set of s -dimensional probability vectors $p = (p_1, \dots, p_s)$ and we write $p \in \Delta_{s-1}$ meaning $(p_1, \dots, p_{s-1}) \in \Delta_{s-1}$.

Let $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and the maximum eigenvalues of a matrix respectively. As before, for a covariance matrix Σ , $Q = \Sigma^{-1}$, $K = \sqrt{Q}$. For probability weights $p = (p_1, \dots, p_s)$, let

$$D_p = \text{diag}(p_1 Q_{11}^{-1}, \dots, p_s Q_{ss}^{-1}) \quad (2.7)$$

be a $d \times d$ block-diagonal matrix.

Definition (Pseudo-spectral gap). For arbitrary distribution π with precision matrix Q , and any probability vector $p \in \Delta_{s-1}$, the pseudo-spectral gap for $\text{RSGS}(p)$ is defined as

$$\text{P-Gap}(p) := 1 - \lambda_{\max}(I - K D_p K), \quad (2.8)$$

which due to Lemma 2 can be written as

$$\text{P-Gap}(p) = 1 - \lambda_{\max}(I - D_p Q) = \lambda_{\min}(D_p Q). \quad (2.9)$$

Weights $p^{\text{opt}} = (p_1^{\text{opt}}, \dots, p_s^{\text{opt}}) \in \Delta_{s-1}$ are called pseudo-optimal for RSGS if they maximise the corresponding pseudo-spectral gap, i.e.,

$$p^{\text{opt}} = \underset{p \in \Delta_{s-1}}{\text{argmax}} \lambda_{\min}(D_p Q). \quad (2.10)$$

Remark. Theorem 1 implies that for $\text{RSGS}(p)$ the pseudo-spectral and the spectral gap are the same if the target distribution is normal.

Useful observation for both theoretical and practical purposes is the uniqueness of the pseudo-optimal weights.

Theorem 3. There exists a unique solution for (2.10).

We conclude this section by presenting an upper bound on the possible improvement of the spectral gap of $\text{RSGS}(p^{\text{opt}})$ compared to the spectral gap of the vanilla chain, i.e., the chain with uniform selection probabilities.

Theorem 4. Let $\text{Gap}(p)$ be the spectral gap of $\text{RSGS}(p)$ and $\text{Gap}(\frac{1}{s})$ be the spectral gap of the vanilla chain, i.e., the RSGS with uniform selection probabilities. Then for any probability vectors p and q

$$\text{Gap}(p) \leq \left(\max_{i=1, \dots, s} \frac{p_i}{q_i} \right) \text{Gap}(q),$$

in particular,

$$\text{Gap}(p) \leq \left(\max_{i=1,\dots,s} sp_i \right) \text{Gap} \left(\frac{1}{s} \right), \quad (2.11)$$

where s is the number of components in the Gibbs sampling scheme.

Remark. Theorem 4 implies

$$\text{P-Gap}(p) \leq \left(\max_{i=1,\dots,s} sp_i \right) \text{P-Gap} \left(\frac{1}{s} \right), \quad (2.12)$$

where $\text{P-Gap}(\frac{1}{s})$ is the pseudo-spectral gap for the vanilla chain.

Theorem 4 states that the maximum gain one can get by using non-uniform selection probabilities is bounded by s times - the number of blocks in the Gibbs sampling scheme. Thus, we expect the pseudo-optimal weights to be particularly useful in high dimensional settings.

2.3 Motivating examples

The pseudo-optimal weights (2.10) have complicated interpretation as we will see in the following examples.

Example 1. In case where the correlation matrix of the target distribution has blocks of highly correlated coordinates, one would prefer to update them more frequently than the others. In this section we construct an artificial example where the upper bound in (2.12) is $\frac{d}{2} \text{Gap} \left(\frac{1}{d} \right)$. Consider a target distribution in \mathbb{R}^d , $d = 2k$ with correlation and normalised precision (inverse covariance) matrices given respectively by their block form, i.e., $\text{Corr} = (C_{ij})_{i,j=1}^k$, $Q = (Q_{ij})_{i,j=1}^k$, where C_{ij} and Q_{ij} are 2×2 matrices such that all Q_{ij} , C_{ij} are zero matrices if $i \neq j$ and for all $i = 1, \dots, k$

$$C_{ii} = \begin{pmatrix} 1 & -\rho_i \\ -\rho_i & 1 \end{pmatrix}, \quad Q_{ii} = \begin{pmatrix} 1 & \rho_i \\ \rho_i & 1 \end{pmatrix},$$

where we assume $\rho_i \geq 0$, $i = 1, \dots, k$. Assume one wants to apply the coordinate-wise RSGS to sample from a distribution with the above correlation matrix.

Proposition 1. *Let the inverse covariance matrix Q be as above. Define*

$$\alpha_i = \frac{\prod_{l=1, l \neq i}^k (1 - \rho_l)}{\sum_{l=1}^k \prod_{j=1, j \neq l}^k (1 - \rho_j)}. \quad (2.13)$$

Then the pseudo-optimal weights are given by

$$p_{2i-1}^{\text{opt}} = p_{2i}^{\text{opt}} = \frac{\alpha_i}{2}. \quad (2.14)$$

The corresponding P-Gap is

$$\text{P-Gap}(p^{\text{opt}}) = \frac{\prod_{l=1}^k (1 - \rho_l)}{2 \sum_{l=1}^k \prod_{j=1, j \neq l}^k (1 - \rho_j)}. \quad (2.15)$$

Without loss of generality assume $\rho_1 = \max\{\rho_1, \dots, \rho_k\}$. We shall compare pseudo-spectral gaps of the vanilla chain with $\text{RSGS}(p^{\text{opt}})$. One can easily obtain that the pseudo-spectral gap of the vanilla chain is given by

$$\text{P-Gap}\left(\frac{1}{d}\right) = \frac{1}{d}(1 - \rho_1).$$

Simple calculations yield

$$\begin{aligned} \lim_{\rho_1 \rightarrow 1} \frac{\text{P-Gap}\left(\frac{1}{d}\right)}{\text{P-Gap}(p^{\text{opt}})} &= \lim_{\rho_1 \rightarrow 1} \frac{1 - \rho_1}{2k \left(\frac{\prod_{l=1}^k (1 - \rho_l)}{2 \sum_{l=1}^k \prod_{j=1, j \neq l}^k (1 - \rho_j)} \right)} \\ &= \lim_{\rho_1 \rightarrow 1} \frac{1}{k} \frac{\left(\sum_{l=1}^k \prod_{j=1, j \neq l}^k (1 - \rho_j) \right)}{\prod_{l=2}^k (1 - \rho_l)} = \frac{1}{k} = \frac{2}{d}. \end{aligned}$$

Moreover,

$$\lim_{\rho_1 \rightarrow 1} \left(\max_i dp_i^{\text{opt}} \right) = \frac{1}{k} = \frac{2}{d}.$$

Thus, we obtained a sequence of precision matrices for which the pseudo-optimal weights improve the pseudo-spectral gap by $\frac{d}{2}$ times in the limit which is the upper bound in (2.12). Notice, if the underlying target distribution is normal, the upper bound in (2.11) for the L_2 -spectral gap is approximated.

Remark. Corollary 1 to Theorem 5 of [Roberts and Sahu \[1997a\]](#) implies that the spectral gap of Deterministic Update Gibbs Sampler (denoted by $\text{Gap}(\text{DUGS})$) for the normal target with a 3-diagonal precision Q is greater than the gap of the vanilla

RSGS (i.e., with the uniform selection probabilities). Moreover, from Corollary 2 to Theorem 5 of [Roberts and Sahu \[1997a\]](#), $\lim_{\rho_1 \rightarrow 1} \frac{\text{Gap}(\text{DUGS})}{\text{P-Gap}(\frac{1}{d})} = 2$. We constructed an example of a 3-diagonal precision matrix, where in dimensions greater than 6, RSGS with pseudo-optimal weights p^{opt} converges $\frac{d}{4}$ times faster than DUGS for $\rho_1 \rightarrow 1$.

Example 2. One mistakenly might conclude that significant gain from using the pseudo-optimal weights is achieved only if some of the off-diagonal entries of the covariance matrix are close to one. Here we provide a somewhat counter-intuitive example that demonstrates fallacy of such statement.

Consider a correlation matrix matrix given by $\Sigma^{(2)} = (C_{ij})_{i,j=1}^d$, where $C_{ii} = 1$ for $i = 1, \dots, d$, $C_{1i} = C_{i1} := c_i \geq 0$ for $i = 2, \dots, d$ and all other entries $C_{ij} = 0$.

One can easily work out that the smallest eigenvalue of $\Sigma^{(2)}$, $\lambda_{\min} = 1 - \sqrt{\sum_{i=2}^d c_i^2}$. Thus, if $\lambda_{\min} > 0$, then $\Sigma^{(2)}$ is a valid correlation matrix. Set $d = 50$ and $c_i = \frac{1}{7.01} \approx 0.143$ for $i = 2, \dots, 50$.

We run the subgradient optimisation algorithm presented in Section 2.4 in order to estimate p^{opt} . We estimate $p_1^{\text{opt}} \approx 0.484$, $p_i^{\text{opt}} \approx 0.01$ for $i = 2, \dots, 50$. From (2.9) the pseudo-spectral gap is roughly $\frac{1}{1496}$, whilst $\text{P-Gap}(\frac{1}{50})$ is roughly $\frac{1}{18294}$. Thus, if the target distribution is normal, the spectral-gap of the vanilla RSGS is improved by more than 12 times. Note, however, all off-diagonal correlations are less than 0.143.

2.4 Adapting Gibbs Sampler

In this section we derive the Adaptive Random Scan Gibbs Sampler (ARSGS) Algorithm 9. We provide all the steps and intuition leading towards the final working version of the algorithm presented in the end of the section.

The goal is to compute the pseudo-optimal weights (2.10) for the RSGS (2.1). However, in practice the correlation matrix of the target distribution is usually not known. Thus, we could proceed in the adaptive way, similarly to Haario et al. [Haario et al. \[2001\]](#). Given output of the chain of length n , let $\hat{\Sigma}_n$, \hat{Q}_n , and $\widehat{D(p)}_n$ be estimators of Σ , Q , and D_p respectively built upon the chain output. For instance, one may choose the naive estimator

$$\hat{\Sigma}_n = \frac{1}{n} \left(\sum_{i=0}^n X_i X_i^T - (n+1) \bar{X}_n \bar{X}_n^T \right), \quad (2.16)$$

where X_n is the chain output at time n and \bar{X}_n is a sample mean of the output up

to time n .

Algorithm 5: Adaptive Random Scan Gibbs Sampler (general idea)

Generate a starting location $X_0 \in \mathbb{R}^d$. Set an initial value of $p^0 \in \Delta_{s-1}$.

Choose a sequence of positive integers $(k_m)_{m=0}^\infty$. Set $n = 0$, $i = 0$.

Beginning of the loop

1. $n := n + k_i$. Run RSGS(p^i) for k_i steps;
2. Re-estimate $\widehat{\Sigma}_n$ and $\widehat{D(p^i)_n}$;
3. Compute $p^{i+1} = \operatorname{argmax}_{p \in \Delta_{s-1}} \lambda_{\min} \left(\widehat{D(p^i)_n} \widehat{Q}_n \right)$;
4. $i := i + 1$.

Go to **Beginning of the loop**

The Algorithm 5 summarises the above ideas. The algorithm is limited by Step 3, where one needs to maximise the minimum eigenvalue. Maximising the minimum eigenvalue is known to be a complicated optimisation problem. There is vast literature covering optimisation problem in Step 3 and we refer to Overton [1988, 1992], Chu [1990], and references therein. Unfortunately, the existing optimisation algorithms require computation of the minimum eigenvalues of $\lambda_{\min} \left(\widehat{D(p)_n} \widehat{Q}_n \right)$, which is an expensive procedure. Since at every iteration of Algorithm 5 the weights p^i are suboptimal, it is reasonable to solve the optimisation problem in Step 3 approximately in order to reduce the computational time of each iteration. Therefore, we develop a new algorithm based on the subgradient method for convex functions (see Chapter 8 of Bertsekas [2003]) applied to (2.10).

For $\varepsilon > 0$, introduce a contraction set of Δ_s :

$$\Delta_s^\varepsilon := \{w \in \mathbb{R}^s \mid w_i \geq \varepsilon, i = 1, \dots, s-1; 1 - w_1 - \dots - w_s \geq \varepsilon\}, \quad (2.17)$$

and consider $(d+1) \times (d+1)$ matrices

$$\begin{aligned} Q^{\text{ext}} &= \operatorname{diag}(Q, 1), & \Sigma^{\text{ext}} &= \operatorname{diag}(\Sigma, 1), \\ \Sigma_n^{\text{ext}} &= \operatorname{diag}(\widehat{\Sigma}_n, 1), & Q_n^{\text{ext}} &= \operatorname{diag}(\widehat{Q}_n, 1), \\ D_w^{\text{ext}} &= \operatorname{diag}\left(D_w, 1 - \sum_{i=1}^s w_i\right), & D_n^{\text{ext}}(w) &= \operatorname{diag}\left(\widehat{D(w)_n}, 1 - \sum_{i=1}^s w_i\right). \end{aligned}$$

Let us denote the target function

$$f(w) = \lambda_{\min} (D_w^{\text{ext}} Q^{\text{ext}}) = \lambda_{\min} \left(\sqrt{Q^{\text{ext}}} D_w^{\text{ext}} \sqrt{Q^{\text{ext}}} \right), \quad (2.18)$$

where the last equality holds in view of Lemma 2.

Using the definition of the pseudo-optimal selection probabilities (2.10), one can easily verify the following proposition

Proposition 2. *The pseudo-optimal weights (2.10) can be obtained as a normalised solution of*

$$w^* = \operatorname{argmax}_{w \in \Delta_s} f(w),$$

i.e.,

$$p_j^{\text{opt}} = \frac{w_j^*}{w_1^* + \dots + w_s^*}, \quad j = 1, \dots, s, \quad (2.19)$$

Moreover,

$$\text{P-Gap}(p^{\text{opt}}) = \frac{1}{(w_1^* + \dots + w_s^*)} f(w^*),$$

where f is defined in (2.18).

Remark. One could easily avoid introducing the extended matrices $\Sigma^{\text{ext}}, Q^{\text{ext}}$ by simply setting $p_s = 1 - p_1 - \dots - p_{s-1}$ and treating function $\lambda_{\min} \left(\widehat{D(p)_n} \widehat{Q_n} \right)$ as a function of $s - 1$ variables. However, we found empirically, that such approach can significantly slow down convergence of the ARSGS Algorithm 9 introduced later in this section.

It is easy to prove concavity of the function f (2.18).

Proposition 3. *Function f defined in (2.18) is concave in Δ_s .*

Andrew et al. [1993] show that f is differentiable at $w \in \Delta_s$ if and only if $f(w)$ is a simple eigenvalue of $\sqrt{Q^{\text{ext}}} D_w^{\text{ext}} \sqrt{Q^{\text{ext}}}$. It is also known that convex functions in Euclidean spaces are differentiable almost everywhere w.r.t. Lebesgue measure (see Borwein and Vanderwerff [2010], Section 2.5). Andrew et al. [1993] also provide exact formulas for computing derivatives of f where they exist. Thus, we are motivated to adapt subgradient method for convex functions in order to modify Step 3 in the above algorithm.

Let $\langle \cdot, \cdot \rangle$ denote scalar product in \mathbb{R}^d . Recall the definition of subgradient and subdifferential.

Definition. Let $h : \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex function. We say v is a subgradient of h at point x if for all $y \in \mathbb{R}^d$,

$$h(y) \geq h(x) + \langle y - x, v \rangle.$$

If h is concave, we say that v is a supergradient of h at a point x , if $(-v)$ is a subgradient of the convex function $(-h)$ at x . The set of all sub-(super-)gradients at the point x is called sub-(super-)differential at x and is denoted by $\partial h(x)$.

In other word, $\partial h(x)$ parametrises a collection of all tangent hyperplanes at a point x .

Note that $f(w) = 0$ on the boundary of Δ_s . Therefore, the maximum of f is attained inside Δ_s . One may apply the subgradient optimisation method in order to estimate p^{opt} . The method is described in Algorithm 6.

Algorithm 6: Subgradient optimisation algorithm

Set an initial value of $w^0 = (w_1^0, \dots, w_s^0) \in \Delta_s$. Define a sequence of non-negative numbers $(a_m)_{m=1}^\infty$ such that $\sum_{m=1}^\infty a_m = \infty$ and $\lim_{m \rightarrow \infty} a_m = 0$. Set $i = 0$.

Beginning of the loop

1. Compute any $d^i \in \partial f(w^i)$. Normalise $d^i := \frac{d^i}{|d_1^i| + \dots + |d_s^i|}$;
2. $w_j^{\text{new}} := w_j^i + a_{i+1} d_j^i$, $j = 1, \dots, s$;
3. $w^{i+1} := \text{Pr}_{\Delta_s}(w^{\text{new}})$, where Pr_{Δ_s} is the projection operator on Δ_s ;
4. $i := i + 1$.

Go to **Beginning of the loop**

It is known that Algorithm 6 produces a sequence $\{w^i\}$ such that $w^i \rightarrow w^*$ as $i \rightarrow \infty$ (see Chapter 8 of Bertsekas [2003]). Therefore, it is reasonable to combine the Algorithm 5 with the subgradient algorithm. In order to do so, define a sequence of approximations of (2.18):

$$f_n(w) = \lambda_{\min}(D_n^{\text{ext}}(w)Q_n^{\text{ext}}) = \lambda_{\min}\left(\sqrt{Q_n^{\text{ext}}}D_n^{\text{ext}}(w)\sqrt{Q_n^{\text{ext}}}\right).$$

Algorithm 7 resembles the aforementioned ideas. Here we consider iterations w^i to be in Δ_s^ε for $\varepsilon > 0$ because of three reasons. Firstly, the RSGS with

Algorithm 7: Adaptive Gibbs Sampler based on subgradient optimisation method (not implementable)

Generate a starting location $X_0 \in \mathbb{R}^d$. Fix $\frac{1}{s+1} > \varepsilon > 0$. Set an initial value of $w^0 = (w_1^0, \dots, w_s^0) \in \Delta_s^\varepsilon$. Define a sequence of non-negative numbers $(a_m)_{m=1}^\infty$ such that $\sum_{m=1}^\infty a_m = \infty$ and $\lim_{m \rightarrow \infty} a_m = 0$. Set $i = 0$. Choose a sequence of positive integers $(k_m)_{m=0}^\infty$.

Beginning of the loop

1. $n := n + k_i$. $p_j^i := \frac{w_j^i}{w_1^i + \dots + w_s^i}$, $j = 1, \dots, s$. Run RSGS(p^i) for k_i steps;
2. Re-estimate $\hat{\Sigma}_n$ and $D_n^{\text{ext}}(w)$;
- 3.1. Compute $d^i \in \partial f_n(w^i)$. Normalise $d^i := \frac{d^i}{|d_1^i| + \dots + |d_s^i|}$;
- 3.2. $w_j^{\text{new}} := w_j^i + a_{i+1} d_j^i$, $j = 1, \dots, s$;
- 3.3. $w^{i+1} := \text{Pr}_{\Delta_s^\varepsilon}(w^{\text{new}})$, where $\text{Pr}_{\Delta_s^\varepsilon}$ is the projection operator on Δ_s^ε ;
4. $i := i + 1$.

Go to **Beginning of the loop**

selection probabilities that are on the boundary of Δ_{s-1} is not ergodic. Secondly, this assumption is motivated by the results of [Latuszyński et al. \[2013b\]](#), where it is a minimum requirement to establish convergence of an Adaptive Gibbs Sampler. Finally, in the final Algorithm 9, it is an essential assumption to be able to perform power iteration Step 3.1.1. Note, however, that $\varepsilon > 0$ may be chosen arbitrary small.

In order to construct an implementable and practical ARSGS algorithm, we still need to find a way to approximate the subgradient $\partial f_n(w^i)$ in Step 3.1 and also find a cheap way of computing the projection $\text{Pr}_{\Delta_s^\varepsilon}(w^{\text{new}})$ in Step 3.3.

An efficient algorithm to compute the projection on Δ_s^ε is presented in [Wang and Carreira-Perpiñán \[2013\]](#) and summarised in Algorithm 8. First, we increase all small coordinates to be ε in Step 1. If the resulting point is outside Δ_s^ε , we need to project it on the hyperplane $\{w \in \mathbb{R}^s | 1 - \sum_{j=1}^s w_j = \varepsilon, w_i \geq \varepsilon\}$. In order to find the projection, we first rescale the coordinates in Step 3. Then we use the algorithm of [Wang and Carreira-Perpiñán \[2013\]](#) to compute the projection on $\{w \in \mathbb{R}^s | \sum_{j=1}^s w_j = 1, w_i \geq 0\}$ in Steps 4 - 6. Finally, we rescale the resulting point in Step 7 and thus, obtain the desired projection.

We are left to construct a procedure that approximates a supergradient $d^i \in$

Algorithm 8: Projection on Δ_s^ε

The output of the algorithm is w^{proj} - projection of $w \in \mathbb{R}^s$ onto Δ_s^ε .

1. Define an auxiliary variable $w^{\text{aux}} := w$. For $j = 1, \dots, s$, if $w_j^{\text{aux}} < \varepsilon$, set $w_j^{\text{aux}} := \varepsilon$;
 2. If $1 - \sum_{j=1}^s w_j^{\text{aux}} > \varepsilon$, then $w^{\text{proj}} := w^{\text{aux}}$ and go to Step 8;
else go to Step 3;
 3. For $j = 1, \dots, s$, $w_j^{\text{temp}} := \frac{1}{1-\varepsilon(s+1)}(w_j^{\text{aux}} - \varepsilon)$;
 4. Sort vector $(w_1^{\text{temp}}, \dots, w_s^{\text{temp}})$ into $u : u_1 \geq \dots \geq u_s$;
 5. $\rho := \max \left\{ 1 \leq j \leq s : u_j + \frac{1}{j} \left(1 - \sum_{k=1}^j u_k \right) > 0 \right\}$;
 6. Define $\lambda = \frac{1}{\rho} \left(1 - \sum_{k=1}^\rho u_k \right)$;
 7. For $j = 1, \dots, s$, $w_j^{\text{proj}} := \varepsilon + (1 - \varepsilon(s+1)) \max\{w_j^{\text{temp}} + \lambda, 0\}$;
 8. Return w^{proj} .
-

$\partial f_n(w^i)$ in Step 3.1 of Algorithm 7. Since $f_n(w)$ is the minimum eigenvalue of a self-adjoint matrix, $f_n(w)$ may be obtained as

$$f_n(w) = \min_{x: \|x\|=1} \left\langle \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x, x \right\rangle,$$

where $x \in \mathbb{R}^{d+1}$ and $\langle \cdot, \cdot \rangle$ denotes scalar product in \mathbb{R}^d . Define

$$g_x^n(w) = \left\langle \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x, x \right\rangle.$$

Let ∇ denote a gradient w.r.t. w . Then

$$\nabla g_x^n(w) = \left(\left\langle \sqrt{Q_n^{\text{ext}}} \frac{\partial D_n^{\text{ext}}(w)}{\partial w_1} \sqrt{Q_n^{\text{ext}}} x, x \right\rangle, \dots, \left\langle \sqrt{Q_n^{\text{ext}}} \frac{\partial D_n^{\text{ext}}(w)}{\partial w_s} \sqrt{Q_n^{\text{ext}}} x, x \right\rangle \right). \quad (2.20)$$

Here $\frac{\partial}{\partial w_i}$ stands for the element-wise derivative w.r.t. w_i , $i = 1, \dots, s$. Ioffe-Tikhomirov theorem (see, e.g., Zălinescu [2002]) implies that the superdifferential of f_n at a point $w \in \Delta_s^\varepsilon$ can be computed as

$$\partial f_n(w) = \text{conv} \left\{ \nabla g_x(w) \mid x : \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x = f_n(w)x, \|x\| = 1 \right\},$$

where $\text{conv}\{A\}$ denotes a convex hull of the set A .

Computing elements of the set $\partial f_n(w)$ is computationally expensive, since one has to calculate the minimum eigenvectors of $\sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}}$. Therefore, we look for a cheap approximation of the points $\nabla g_x^n(w)$ in $\partial f_n(w)$.

Let $y = \sqrt{Q_n^{\text{ext}}} x$. Since we are interested in minimum eigenvectors x , such that

$$\sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x = f_n(w)x,$$

we can rewrite this equation as

$$\frac{1}{f_n(w)} y = (D_n^{\text{ext}}(w))^{-1} \Sigma_n^{\text{ext}} y. \quad (2.21)$$

That is, computing the minimum eigenvector of $\sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}}$ is equivalent to computing the maximum eigenvector of $(D_n^{\text{ext}}(w))^{-1} \hat{\Sigma}_n$. Given y that solves (2.21) and substituting $x = \frac{1}{\|\sqrt{\Sigma_n^{\text{ext}}} y\|} \sqrt{\Sigma_n^{\text{ext}}} y$ into (2.20), we obtain

$$\nabla g_x^n(w) = \frac{1}{\|\sqrt{\Sigma_n^{\text{ext}}} y\|^2} \left(\left\langle \frac{\partial D_n^{\text{ext}}(w)}{\partial w_1} y, y \right\rangle, \dots, \left\langle \frac{\partial D_n^{\text{ext}}(w)}{\partial w_s} y, y \right\rangle \right). \quad (2.22)$$

We can do further transformations. Let

$$(D_n^{\text{ext}}(w))^{-1} = L_n(w) L_n^T(w) \quad (2.23)$$

be the Cholesky decomposition of $(D_n^{\text{ext}}(w))^{-1}$, where $L_n(w)$ is a lower triangular matrix. Define $z := L_n^{-1}(w)y$ and

$$R_i(w) := \text{diag} \left(0, \dots, 0, \frac{1}{w_i}, \dots, \frac{1}{w_i}, 0, \dots, 0, -\frac{1}{1 - w_1 - \dots - w_s} \right), \quad (2.24)$$

where $\frac{1}{w_i}$ are placed exactly on the positions of the diagonal elements of Q_{ii} in the partition $Q = (Q_{ij})_{i,j=1}^s$. Then after simple manipulations, (2.21) and (2.22) are equivalent respectively to

$$L_n^T(w) \Sigma_n^{\text{ext}} L_n(w) z = \frac{1}{f_n(w)} z$$

and

$$\nabla g_x^n(w) = \frac{1}{\langle L_n^T(w) \Sigma_n^{\text{ext}} L_n(w) z, z \rangle} \begin{pmatrix} \langle R_1(p) z, z \rangle, \dots, \langle R_{s-1}(p) z, z \rangle \end{pmatrix}, \quad (2.25)$$

where we used the block-diagonal structure of $L_n(w)$ and a representation

$$\frac{\partial D_n^{\text{ext}}(w)}{\partial w_i} = \text{diag}(0, \dots, 0, Q_{ii}^{-1}, 0, \dots, 0, -1).$$

Because of the normalisation in Step 3.1 of the Adaptive Gibbs Sampler 7, (2.22) and (2.25) imply that a supergradient of $f_n(w)$ is proportional to

$$d_y(w) = (\langle (D_n^{\text{ext}}(w))_{11} y, y \rangle, \dots, \langle (D_n^{\text{ext}}(w))_{s-1} y, y \rangle), \quad (2.26)$$

or, in terms of z , to

$$d_z(w) = (\langle R_1 z, z \rangle, \dots, \langle R_s z, z \rangle), \quad (2.27)$$

where y and z are the maximum eigenvectors of $(D_n^{\text{ext}}(w))^{-1} \hat{\Sigma}_n$ and $L_n^T(w) \Sigma_n^{\text{ext}} L_n(w)$, respectively. Here the lower triangular matrix $L_n(w)$ is defined by the Cholesky decomposition (2.23).

Power iteration step may be performed in order to approximate y and z . Let z_0 and y_0 be randomly generated unit vectors. Then at every iteration of the algorithm, we compute

$$y_{i+1} = Q_n^{\text{ext}} D_n^{\text{ext}}(w^i) y_i, \quad (2.28)$$

$$z_{i+1} = L_n^T(w^i) \Sigma_n^{\text{ext}} L_n(w^i) z_i. \quad (2.29)$$

and use the normalised vectors y_{i+1} and z_{i+1} instead of y and z when computing the directions (2.26) and (2.27)

Given the intuition above, we present two versions of the ARSGS in the Algorithm 9, where in round brackets we denote an alternative version of the algorithm.

One might notice the perturbation term $b_{i+1} \xi_{i+1}$ in the Step 3.1.1. In fact, without the perturbation we may break the algorithm due to the fact that the power iteration step may fail to approach the maximum eigenvalue. It happens when z_i (or y_i) "slips" into the eigenspace of a wrong eigenvalue and can't get out of it for the subsequent algorithm steps.

The simplest example one can think of is sampling from $N(0, I_2)$ using

coordinate-wise RSGS. Set $\Sigma_n^{\text{ext}} = I_3$. If one starts from $w^0 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{2})$ and steps a_m (see Algorithm 9 for the meaning of a_m) are chosen to be tiny, $(0, 0, 1)$ is the maximum eigenvector of $L_n^T(w^i)\Sigma_n^{\text{ext}}L_n(w^i) = \text{diag}(w_1^i, w_2^i, 1 - w_1^i - w_2^i)$ for $i = 1, \dots, N_0$, where N_0 depends on the sequence a_m . If N_0 is big enough (equivalently, a_m is small enough), eventually $z_i = (0, 0, 1)$ for all i due to the computational precision error and we will not get out of this eigenspace. Therefore, there is a possibility that eventually w^i sticks to the boundary of Δ_s^ε . To surpass the issue we modify the power iteration Step 3.1.1 by perturbing the values of z_i ,

$$z_{i+1} = L_n^T(w^i)\Sigma_n^{\text{ext}}L_n(w^i)z_i + b_i\xi_i,$$

where b_i is a non-negative sequence convergent to 0, and ξ_i is i.i.d. sequence of points uniformly distributed on the unit sphere.

Remark. In Step 3.1.1 of the ARSGS Algorithm 9, $\frac{1}{\|L_n^T(w^i)\Sigma_n^{\text{ext}}L_n(w^i)z_i\|}$ and $\frac{1}{\|(D_n^{\text{ext}}(w^i))^{-1}\Sigma_n y_i\|}$ are approximations of $\max_{w \in \Delta_s^\varepsilon} f(w)$, where f is defined in (2.18). Therefore, taking into an account Proposition 2, we can estimate P-Gap(p^{opt}) by

$$\text{P-Gap}(p^{\text{opt}}) \approx ((w_1^i + \dots + w_s^i) \|L_n^T(w^i)\Sigma_n^{\text{ext}}L_n(w^i)z_i\|)^{-1} \quad (2.30)$$

or

$$\text{P-Gap}(p^{\text{opt}}) \approx \left((w_1^i + \dots + w_s^i) \| (D_n^{\text{ext}}(w^i))^{-1} \Sigma_n^{\text{ext}} y_i \| \right)^{-1}. \quad (2.31)$$

2.5 Adapting Metropolis-within-Gibbs

Sometimes one can not or does not want to sample from the full conditionals Pr_i of the target distribution. In this case one may want to proceed with the Metropolis-within-Gibbs algorithm. For simplicity, we restrict ourselves to the coordinate-wise update Random Walk Metropolis-within-Gibbs (RWMwG) Algorithm 10, though the idea presented below goes beyond this particular case.

One should not get confused with the parameter q in Step 2 of the Algorithm 10. If $q = 1$, one recovers the RWMwG algorithm in its canonical form.

It is often not clear how to choose proposal variances β_i to speed up the convergence. We follow Gelman et al. [1996] suggestion that the average acceptance rate α should be around 0.44 and adapt β_i on the fly to keep up with this acceptance rate. Algorithm 11 is the adaptive version of the RWMwG as suggested in

Algorithm 9: Adaptive Random Scan Gibbs Sampler (final version)

Generate a starting location $X_0 \in \mathbb{R}^d$. Fix $\frac{1}{s+1} > \varepsilon > 0$. Set an initial value of $w^0 = (w_1^0, \dots, w_s^0) \in \Delta_s^\varepsilon$, generate a random unit vectors $z_0 \in \mathbb{R}^{d+1}$ (or $y_0 \in \mathbb{R}^{d+1}$). Define two sequences of non-negative numbers $(b_m)_{m=1}^\infty$ and $(a_m)_{m=1}^\infty$ such that $\sum_{m=1}^\infty a_m = \infty$, $a_m \rightarrow 0$ and $b_m \rightarrow 0$ as $m \rightarrow \infty$. Set $i = 0$. Choose a sequence of positive integers $(k_m)_{m=0}^\infty$.

Beginning of the loop

1. $n := n + k_i$. $p_j^i := \frac{w_j^i}{w_1^i + \dots + w_s^i}$, $j = 1, \dots, s$. Run RSGS(p^i) for k_i steps;

2. Re-estimate $\hat{\Sigma}_n$. Recompute Σ_n^{ext} , $D_n^{\text{ext}}(w^i)$, $L_n(w^i)$;

3.1. Compute approximate gradient direction d^i :

3.1.1. Generate $\xi_{i+1} \sim N(0, I_{d+1})$. $\xi_{i+1} := \frac{\xi_{i+1}}{\|\xi_{i+1}\|}$.
Compute

$$z_{i+1} := L_n^T(w^i) \Sigma_n^{\text{ext}} L_n(w^i) z_i + b_{i+1} \xi_{i+1}.$$

$$\left(y_{i+1} := (D_n^{\text{ext}}(w^i))^{-1} \Sigma_n^{\text{ext}} y_i + b_{i+1} \xi_{i+1} \right)$$

$$\text{Normalise } z_{i+1} := \frac{z_{i+1}}{\|z_{i+1}\|} \cdot \left(y_{i+1} := \frac{y_{i+1}}{\|y_{i+1}\|} \right).$$

3.1.2. Compute $d^i = d_{z_{i+1}}(w^i)$ from (2.26) $\left(d^i = d_{y_{i+1}}(w^i) \text{ from (2.27)} \right)$.

$$\text{Normalise } d^i := \frac{d^i}{|d_1^i| + \dots + |d_s^i|};$$

3.2. $w_j^{\text{new}} := w_j^i + a_{i+1} d_j^i$, $j = 1, \dots, s$;

3.3. Using Algorithm 8 compute projection w^{i+1} of w^{new} onto Δ_s^ε ;

4. $i := i + 1$.

Go to **Beginning of the loop**

Latuszyński et al. [2013b].

One could also adapt the selection probabilities p_i but, as noted in Latuszyński et al. [2013b], there is no to-date guidance on the optimal choice of p_i . Heuristically, we would expect the Adaptive RWMwG to mimic the RSGS, so that we find it to be reasonable to adapt the selection probabilities p_i in the same manner as for the RSGS. Therefore, we introduce Adaptive Random Walk Metropolis within Adaptive Gibbs (ARWMwAG) Sampler described in Algorithm 12, where running the ARWMwG sampler in Step 1 alternates with adaptation of the selection probabilities in Step 2.

Algorithm 10: Random Walk Metropolis-within-Gibbs

Generate a starting location $X_0 \in \mathbb{R}^d$. Let (p_1, \dots, p_s) be a probability vector, $0 < q \leq 1$, $\sigma^2 > 0$. Fix variances β_1, \dots, β_s and choose starting location $(X_1^0, \dots, X_d^0) \in \mathbb{R}^d$. $n := 0$.

Beginning of the loop

1. Sample $i \in \{1, \dots, s\}$ from probability distribution (p_1, \dots, p_s) ;

$$2. \text{ Draw } Y \sim \begin{cases} N(X_i, \beta_i^2) & \text{with probability } q, \\ N(X_i, \sigma^2) & \text{with probability } 1 - q; \end{cases}$$

3. Compute acceptance rate $\alpha = \min \left\{ 1, \frac{\pi(Y|X_{-i}^n)}{\pi(X_i^n|X_{-i}^n)} \right\}$;

4. With probability α accept the proposal and set

$$X^{n+1} = (X_1^n, \dots, X_{i-1}^n, Y, X_{i+1}^n, \dots, X_s^n),$$

otherwise, reject the proposal and set $X^{n+1} = X^n$;

5. $n = n + 1$.

Go to **Beginning of the loop**

Algorithm 11: Adaptive Random Walk Metropolis-within-Gibbs

Generate a starting location $X_0 \in \mathbb{R}^d$. Let (p_1, \dots, p_s) be a probability vector, $0 < q \leq 1$, $\sigma^2 > 0$. Fix variances $\beta_1^0, \dots, \beta_s^0$ and choose starting location $(X_1^0, \dots, X_d^0) \in \mathbb{R}^d$. $n := 0$.

Beginning of the loop

1. Do Steps 1 - 4 of RWMwG Algorithm 10 with proposal variances $(\beta_1^n, \dots, \beta_s^n)$;
2. $\beta_i^{n+1} = \beta_i^n \cdot \exp\left(\frac{1}{n^{0.7}}(\alpha - 0.44)\right)$, where α is the acceptance rate in Step 3 of RWMwG Algorithm 10;
3. $n = n + 1$.

Go to **Beginning of the loop**

Algorithm 12: Adaptive Random Walk Metropolis within Adaptive Gibbs

Generate a starting location $X_0 \in \mathbb{R}^d$. Fix variances $\beta_1^0, \dots, \beta_s^0$, $0 < q \leq 1$, and $\sigma^2 > 0$. Choose also $\frac{1}{s+1} > \varepsilon > 0$. Set an initial value of $w^0 = (w_1^0, \dots, w_s^0) \in \Delta_s^\varepsilon$, generate a random unit vector $z_0 \in \mathbb{R}^{d+1}$ (or $y_0 \in \mathbb{R}^{d+1}$). Define two sequences of non-negative numbers $(b_m)_{m=1}^\infty$ and $(a_m)_{m=1}^\infty$ such that $\sum_{m=1}^\infty a_m = \infty$, $a_m \rightarrow 0$ and $b_m \rightarrow 0$ as $m \rightarrow \infty$. Set $i = 0$. Choose a sequence of positive integers $(k_m)_{m=0}^\infty$.

Beginning of the loop

1. $n := n + k_i$. $p_j^i := \frac{w_j^i}{w_1^i + \dots + w_s^i}$, $j = 1, \dots, s$. Iterate k_i times Steps 1 and 2 of ARWMwG Algorithm 11 with sampling weights (p_1^i, \dots, p_s^i, s) and proposal variances $(\beta_1^n, \dots, \beta_s^n)$;
2. Do steps 2 - 4 of ARSGS Algorithm 9.

Go to **Beginning of the loop**

2.6 Ergodicity of the Adaptive Gibbs Sampler

Here $\{P_\gamma\}_{\gamma \in \Gamma}$ is a collection of Markov kernels with a common stationary distribution π . For example, this can be a collection of RSGS kernels (2.1) or the kernels of the Random Walk Metropolis Algorithm 10.

The main result is presented in Theorem 8, where ergodicity of the modified ARSGS Algorithm 14 is established under the *local simultaneous geometric drift* condition (A3). We shall show in Theorem 5 that the local simultaneous geometric

drift is a natural condition for the ARSGS to have. More generally, if the condition **(A3)** holds, we prove ergodicity for a class of modified AMCMC Algorithms **13** in Theorem **7**.

Ergodicity of the ARWMwG and ARWMwAG (Algorithms **11** and **12**) is established under various conditions on the tails of the target distribution π in Section 5 of Łatuszyński et al. [2013b]. In order to fulfil these conditions we, for example, could take arbitrary $0 \leq q < 1$ and large enough σ^2 in the settings of the adaptive algorithms (see Theorems 5.6, 5.9 and Remark 5.8 in Łatuszyński et al. [2013b]).

One can easily see that **(C1)** holds for the ARSGS since the adaptation rate a_m in the Step **3.2** of Algorithm **9** decays to zero.

Verifying the containment condition **(C2)** is less so trivial. In Theorem 3 of Bai et al. [2011] the containment is established if the *simultaneous geometric drift conditions* hold, i.e., if the following assumptions are fulfilled:

- (A0)** *Uniform small set.* There exist a uniform (ν_γ, m) –small set C , i.e., there exists a measurable set $C \in \mathbb{R}^d$, an integer $m \geq 1$, a constant $\delta > 0$ and a probability measure ν_γ probably depending on $\gamma \in \Gamma$, such that

$$P_\gamma^m(x, \cdot) \geq \delta \nu_\gamma(\cdot), \quad x \in C. \quad (2.32)$$

- (A1)** *Simultaneous geometric drift.* There exist numbers $b < \infty$, $0 < \lambda < 1$, and a function $1 \leq V < \infty$, such that $\sup_{x \in C} V(x) < \infty$ and for all $\gamma \in \Gamma$,

$$P_\gamma V \leq \lambda V + b I_C,$$

where $P_\gamma V(x) = \int_{\mathbb{R}^d} V(y) P_\gamma(x, dy)$ and the small set C is defined in **(A0)**.

Where the entire state space is small (i.e., $C = \mathbb{R}^d$ in **(A0)**) for some RSGS kernel P_p , $p \in \Delta_{s-1}^\varepsilon$, ergodicity of the ARSGS is established in Section 4 of Łatuszyński et al. [2013b] (under additional π –irreducibility and aperiodicity assumptions).

In general, one could establish the geometric drift condition **(A1)** and use Theorem 5.1 of Łatuszyński et al. [2013b] to derive the ergodicity. For ARSGS it might be hard to find a drift function that satisfies **(A1)**. Nevertheless, we will now show that the *local simultaneous geometric drift condition* holds given that P_p is geometrically ergodic for some $p \in \Delta_{s-1}^\varepsilon$.

(A2) Geometric ergodicity. There exists $\gamma \in \Gamma$ such that P_γ is geometrically ergodic. That is, P_γ is π -irreducible, aperiodic (see Section 3.2 of [Roberts and Rosenthal \[2004\]](#) for definitions), and there exist drift coefficients $(\lambda_\gamma, V_\gamma, b_\gamma, C_\gamma)$ such that

$$P_\gamma V_\gamma \leq \lambda_\gamma V_\gamma + b_\gamma I_{\{C_\gamma\}}, \quad (2.33)$$

where $b_\gamma < \infty$, $0 \leq \lambda_\gamma < 1$, V_γ is a function such that π -almost surely $1 \leq V_\gamma < \infty$, and $I_{\{C_\gamma\}}$ is an indicator function of a small set C_γ (that is, for all $x \in C_\gamma$, (2.32) holds).

(A3) Local simultaneous geometric drift. For every $\gamma \in \Gamma$, there exists a measurable function $1 \leq V_\gamma < \infty$, a small set C_γ and an open neighbourhood B_γ such that

- (a) C_γ is a uniform small set for $\hat{\gamma} \in B_\gamma$, i.e., (2.32) holds for all $\hat{\gamma} \in B_\gamma$ and $x \in C_\gamma$;
- (b) for all $\hat{\gamma} \in B_\gamma$,

$$P_{\hat{\gamma}} V_\gamma \leq \tilde{\lambda}_\gamma V_\gamma + \tilde{b}_\gamma I_{\{C_\gamma\}} \quad (2.34)$$

for some $\tilde{b}_\gamma < \infty$ and $\tilde{\lambda}_\gamma < 1$.

Theorem 5. Assume **(A2)** for the RSGS kernels P_p , $p \in \Delta_{s-1}^\varepsilon = \Gamma$. Then P_p is geometrically ergodic for each $p \in \Delta_{s-1}^\varepsilon$ and satisfies the local simultaneous drift condition **(A3)**.

Proof of Theorem 5. Since for reversible π -irreducible chains, geometric ergodicity and existence of L_2 -spectral gap are equivalent (see Theorem 2 of [Roberts and Tweedie \[2001\]](#)), the first statement follows from Theorem 4.

Let $(\lambda_p, V_p, b_p, C_p)$ be the drift conditions that satisfy (2.33). For every selection probability vector $p = (p_1, \dots, p_s)$ let $m = m(p) = \min_{i \in \{1, \dots, s-1\}} p_i$. Define norm $|p| = \max_{i \in \{1, \dots, s-1\}} |p_i|$ and take $\delta = \delta(p) > 0$ such that $(1 + (s-1)\delta)\lambda_p \leq 1$. Set $\tilde{\lambda}_p = (1 + (s-1)\delta)\lambda_p$. Then for every \hat{p} such that $|\hat{p} - p| \leq m\delta$,

$$\begin{aligned} P_{\hat{p}} V_p &= \sum_{i=1}^s \hat{p}_i P_{r_i} V_p \leq \sum_{i=1}^s (p_i + (s-1)m\delta) P_{r_i} V_p \leq (1 + (s-1)\delta) \sum_{i=1}^s p_i P_{r_i} V_p \\ &= (1 + (s-1)\delta) P_p V_p \leq \tilde{\lambda}_p V_p + (1 + (s-1)\delta) b_p I_{\{C_p\}}, \end{aligned}$$

where we used representation (2.1) for P_p and the bound

$$|p_s - \hat{p}_{s-1}| \leq \sum_{i=1}^{s-1} |p_i - \hat{p}_i| \leq (s-1)m\delta.$$

We are left to show that the condition (a) of (A3) is satisfied. Indeed, fix any probability vector $p \in \Delta_{s-1}^\varepsilon$. Since C_p is a small set, $P_p^m(x, \cdot) \geq \delta_0 \nu(\cdot)$ for some $m \geq 1$, $\delta_0 > 0$, some probability measure ν and all $x \in C_p$. Then for all $\hat{p} \in \Delta_{s-1}^\varepsilon$,

$$P_{\hat{p}}^m(x, \cdot) \geq \left(\frac{\varepsilon}{\max_{i \in \{1, \dots, s\}} p_i} \right)^m P_p^m(x, \cdot) \geq \left(\frac{\varepsilon}{\max_{i \in \{1, \dots, s\}} p_i} \right)^m \delta_0 \nu(\cdot),$$

whence the condition (a) follows.

□

In order to derive the ergodicity of the ARSGS, we will need the following crucial consequence of the assumption (A3).

Theorem 6. *Assume that Γ is compact in some topology and that the collection of Markov kernels P_γ satisfy (A3). Then there exists a finite partition of Γ into k sets F_i such that*

$$\cup_{i=1}^k F_i = \Gamma,$$

and simultaneous geometric drift conditions (A0) and (A1) hold inside F_i with coefficients (λ, V_i, b, C_i) , where $0 \leq \lambda < 1$, $1 \leq V_i < \infty$, π -a.s., $b < \infty$, and C_i is the uniform small set for $\gamma \in F_i$.

Proof of Theorem 6. Notice,

$$\Gamma \subset \cup_{\gamma \in \Gamma} B_\gamma,$$

where B_γ is an open neighbourhood of γ as in the assumption (A3). Since every open coverage of a compact set Γ has a finite subcover (see, e.g., Theorem 6.37 of Hewitt and Stromberg [1965]), there exist a finite number of B_γ that cover Γ , say $B_{\gamma_1}, \dots, B_{\gamma_k}$. Then one can take $\lambda := \max\{\lambda_{\gamma_1}, \dots, \lambda_{\gamma_k}\}$, $F_i := B_{\gamma_i}$, $b = \max\{b_{\gamma_1}, \dots, b_{\gamma_k}\}$, $C_i := C_{\gamma_i}$.

□

(A4) Assumption (A3) holds and for a chosen set $B \in \mathbb{R}^d$, and the corresponding drift functions $V_\gamma, \gamma \in \Gamma$ are bounded on B , i.e.,

$$\sup_{x \in B} V_\gamma(x) < \infty.$$

We are now ready to state the main ergodicity result.

Theorem 7. *Fix a measurable set $B \subset \mathcal{X}$. Assume Γ is compact in some topology and let $P_\gamma, \gamma \in \Gamma$ be a collection of π -irreducible, aperiodic Markov kernels with a common stationary distribution π . Consider an AMCMC Algorithm 13, where the adaptations are allowed to take place only when the adaptive chain $\{X_n\}$ visits B . Let the conditions (C1), (A3) and (A4) hold and assume that for a starting location (X_0, γ_0) of the adaptive chain, $\mathbb{E}_{(X_0, \gamma_0)} V_{\gamma_0}(X_0) < \infty$, where V_{γ_0} is the drift function for the initial kernel P_{γ_0} . Then the adaptive chain $\{X_n\}$ produced by the Algorithm 13 is ergodic.*

Algorithm 13: Modified AMCMC

Set some initial values for $X_0 \in \mathcal{X}; \gamma_0 \in \Gamma; \bar{\gamma} := \gamma_0; k := 1; n := 0$. Fix any measurable set $B \subset \mathcal{X}$.

Beginning of the loop

1. sample $X_{n+1} \sim P_{\bar{\gamma}}(X_n, \cdot)$;
2. given $\{X_0, \dots, X_{n+1}, \gamma_0, \dots, \gamma_n\}$ update γ_{n+1} according to some adaptation rule;
3. If $X_{n+1} \in B, \bar{\gamma} := \gamma_{n+1}$.

Go to **Beginning of the loop**

Proof of Theorem 7. Since we assume the diminishing adaptation condition (C1), the proof follows once we establish the containment (C2).

Theorem 6 yields there exists a finite partition $\{F_i\}_{i=1}^k$ such that

$$\cup_{i=1}^k F_i = \Gamma,$$

where simultaneous geometric drift conditions hold within every F_i with some drift coefficients (λ, V_i, b, C_i) (as in Theorem 6).

On Γ define a function r such that $r(p) = j$ if $\gamma \in F_j$. (A4) yields there exists $M < \infty$ such that $V_i(x) < M$ for $x \in B, i = 1, \dots, k$.

As in the proof of Theorem 3 of Bai et al. [2011], to verify the containment

condition, it suffices to prove that

$$\sup_n \mathbb{E}[V_{r(\gamma_n)}(X_n)] < \infty,$$

where hereafter $\mathbb{E} = \mathbb{E}_{(X_0, \gamma_0)}$ is the expectation with respect to the probability measure generated by the adaptive chain started from (X_0, p^0) .

Drift condition (2.34) implies

$$\begin{aligned} & \mathbb{E} [V_{r(\gamma_{n+1})}(X_{n+1}) | X_n = x, \gamma_n = \gamma] \\ &= \mathbb{E} [V_{r(\gamma_{n+1})}(X_{n+1}) I_{\{X_{n+1} \notin B\}} | X_n = x, \gamma_n = \gamma] \\ &+ \mathbb{E} [V_{r(\gamma_{n+1})}(X_{n+1}) I_{\{X_{n+1} \in B\}} | X_n = x, \gamma_n = \gamma] \\ &\leq \mathbb{E} [V_{r(\gamma_{n+1})}(X_{n+1}) I_{\{X_{n+1} \notin B\}} | X_n = x, \gamma_n = \gamma] + M \\ &= \mathbb{E} [V_{r(\gamma_n)}(X_{n+1}) I_{\{X_{n+1} \notin B\}} | X_n = x, \gamma_n = \gamma] + M \\ &\leq \mathbb{E} [V_{r(\gamma)}(X_{n+1}) | X_n = x, \gamma_n = \gamma] + M \\ &= P_{r(\gamma)} V_{r(\gamma)}(x) + M \leq \lambda V_{r(\gamma)}(x) + b + M, \end{aligned}$$

where in the first inequality we used the condition (A4) and in the last one we used the fact that $\gamma_{n+1} = \gamma_n$, if $X_{n+1} \notin B$. Here $0 < \lambda < 1$ and $b < \infty$ are as in Theorem 6. Integrating out γ_n and X_n leads to

$$\mathbb{E} [V_{r(\gamma_{n+1})}(X_{n+1})] \leq \lambda \mathbb{E} [V_{r(\gamma_n)}(X_n)] + b + M,$$

implying (see Lemma 2 of Roberts and Rosenthal [2007]),

$$\sup_n \mathbb{E}[V_{r(\gamma_n)}(X_n)] \leq \max \left\{ \mathbb{E} V_{r(\gamma_0)}(X_0), \frac{b + M}{1 - \lambda} \right\} < \infty.$$

□

The reader can easily see that Theorem 7 can be applied to a modified version of the ARSGS Algorithm 14, where the adaptations are allowed to happen only when the adaptive chain hits a set B that satisfies (A4).

Theorem 8. *Fix a measurable set $B \in \mathbb{R}^d$. Consider an Adaptive Random Scan Gibbs Sampler (ARSGS) Algorithm 14 that produces a chain $\{X_n\}$ for which the selection probabilities p^{n+1} are allowed to be changed only if $X_{n+1} \in B$ (i.e., $p^{n+1} = p^n$ if $X_{n+1} \notin B$).*

Let the assumption **(A2)** hold. Then the assumption **(A3)** holds.

Let also **(A4)** be satisfied for the set B and assume that for the starting location (X_0, p^0) of the adaptive chain, $\mathbb{E}_{(X_0, p^0)} V_{p^0}(X_0) < \infty$, where V_{p^0} is the drift function for the initial kernel P_{p^0} . Then the adaptive chain $\{X_n\}$ produced by the ARSGS Algorithm 14 is ergodic.

Proof of Theorem 8. Since Δ_{s-1}^ε is closed and bounded, it is compact (see Heine-Borel Theorem in Hewitt and Stromberg [1965]). Theorem 5 implies that **(A3)** holds. The diminishing adaptation condition **(C1)** holds since $\max_{i \in \{1, \dots, s\}} |p_i^{n+1} - p_i^n| \rightarrow 0$ as $n \rightarrow \infty$, by the construction of the ARSGS Algorithm 9. Therefore, we are in a position to apply Theorem 7 to derive the desired ergodicity of the adaptive chain. \square

Remarks

- 1) We do not have a proof that the ARSGS Algorithm 9 presented in Section 2.4 is ergodic. However, the modified Algorithm 14 is ergodic under the assumptions of Theorem 8. The only difference of the ergodic modification from the original version of the ARSGS is that we do not change the sampling weights p_i if the chain is not in the set B .
- 2) The idea of introducing the set B to an adaptive algorithm comes from the work of Craiu et al. Craiu et al. [2015], where the authors study stability properties (e.g., recurrence) of adaptive chains where the adaptations are allowed to occur only in the set B .
- 3) Assumption **(A4)** is satisfied for level sets $B = B(N) = \cap_{i=1}^k \{x : V_i(x) < N\}$, where V_i are the drift functions as in the proof of the theorem. Theorem 14.2.5. of Meyn and Tweedie [2009] implies that for large N , $B(N)$ covers most of the support of π , meaning that the adaptation will occur in most of the iterations of the Algorithm 14.
- 4) In practice often one can choose B to be any bounded set in \mathbb{R}^d .

2.7 Simulations

It is known that for reversible Markov chains existence of the spectral gap is equivalent to geometric ergodicity (see Theorem 2 of Roberts and Tweedie [2001]). Moreover, geometric ergodicity implies that the Central Limit Theorem holds (see, e.g.,

Algorithm 14: Adaptive Random Scan Gibbs Sampler (ergodic modification)

Generate a starting location $X_0 \in \mathbb{R}^d$. Fix a measurable set $B \subset \mathbb{R}^d$. Fix $\frac{1}{s+1} > \varepsilon > 0$. Set an initial value of $w^0 = (w_1^0, \dots, w_s^0) \in \Delta_s^\varepsilon$, generate random unit vectors $z_0, y_0 \in \mathbb{R}^{d+1}$. Define two sequences of non-negative numbers $(b_m)_{m=1}^\infty$ and $(a_m)_{m=1}^\infty$ such that $\sum_{m=1}^\infty a_m = \infty$, $a_m \rightarrow 0$ and $b_m \rightarrow 0$ as $m \rightarrow \infty$. Set $i = 0$. Choose a sequence of positive integers $(k_m)_{m=0}^\infty$.

Beginning of the loop

1. $n := n + k_i$. Run RSGS(p^i) for k_i steps;
2. Do Steps 2 - 4 of ARSGS Algorithm 9.
3. If current state of chain $X_n \in B$, then $p^i := \frac{w^i}{w_1^i + \dots + w_s^i}$;
Otherwise, $p^i := p^{i-1}$.

Go to **Beginning of the loop**

Bednorz et al. [2008]) . The following theorem of Kipnis & Varadhan Kipnis and Varadhan [1986] states an important relation between the asymptotic variance in CLT and the spectral gap.

Theorem 9. *Assume that P_p is a RSGS kernel (2.1). Then the following upper bound holds, connecting notions of the asymptotic variance with the spectral gap:*

$$\sigma_{as}^2(f) \leq \frac{2 - \text{Gap}(p)}{\text{Gap}(p)} \text{Var}_\pi(f), \quad (2.35)$$

where Var_π denotes variance w.r.t. π and Gap is the spectral gap of P_p . Moreover, if the spectrum of P_p is discrete, then the equality in (2.35) is attained on a second largest eigenfunction of P_p .

Theorem 9 states that by increasing the spectral gap, one decreases the worst case asymptotic variance. Theorem 2 states that the second largest eigenfunction of the RSGS kernel for the normal target distribution is a linear function. Of course, for arbitrary distribution Theorem 2 is false. Nevertheless, we believe that comparing the maximum asymptotic variance over linear functions for the adaptive and non-adaptive algorithms is a reasonable thing to do. Define

$$l_i = l_i(x) = \frac{x_i}{\sqrt{\text{Var}_\pi(x_i)}} \quad (2.36)$$

to be normalised linear functions depending on one coordinate only.

We compute the maximum asymptotic variance in CLT, $\max_{i=1,\dots,d} \sigma_{as}^2(l_i)$, for the adaptive and vanilla RSGS. We believe that in many situations the ratio between the estimated pseudo-spectral gaps is close to the ratio of the maximum asymptotic variances over l_i as follows from Theorem 9. We study also how the pseudo-optimal weights affect the *autocorrelation function* (ACF) of l_i .

Three different examples are studied, where we implement coordinate-wise ARSGS, ARWMwAG and their non-adaptive versions. Two of the examples are in moderate dimension 50: sampling from the posterior in a Poisson Hierarchical Model (PHM) and sampling from the Truncated Multivariate Normal (TMVN) distribution. We also consider sampling from a posterior in a Markov Switching Model (MSM) in 200-dimensional space.

All the asymptotic variances are obtained using the batch-means estimator (see Jones et al. [2006] and Bednorz and Łatuszyński [2007]). Below we outline settings for every problem.

Poisson Hierarchical Model

Gibbs Sampler arises naturally for Hierarchical Models, where our goal is to sample from a posterior distribution. In the present model data Y_i comes from the Poisson distribution with intensity λ_i :

$$Y_i \sim \text{Poisson}(\lambda_i), i = 1, \dots, n, \quad (2.37)$$

where

$$\lambda_i = \exp \left(\sum_{j=1}^d x_{ij} \beta_j \right), \quad (2.38)$$

with $\beta = (\beta_1, \dots, \beta_d)$ being the parameter of interest. Stress that here d is the dimensionality of the problem and n is the number of observations. We set $d = 50$, $n = 100$.

Gibbs sampling through the adaptive rejection sampling presented by Gilks & Wild in Gilks and Wild [1992] is utilised for this problem. See Doss and Narasimhan [1994] for details and formulas for the full conditionals.

We fix the true parameter $\beta_0 := (1, \dots, 1)$ and take the prior distribution on β to be normal with mean $(-1, \dots, -1)$ and variance matrix I_d . We consider two different examples of the design matrix $X = (x_{ij})_{i=1}^n_{j=1}^d$.

Design matrix $X^{(1)}$ is formed as follows. First, we set all the elements to be zero. Let $k = \frac{n}{d} = 2$. Then, we form two upper blocks of ones: $X_{ij}^{(1)} = 1$ for $i \in \{1, \dots, 2k\}$, $j \in \{1, 2\}$, and $X_{ij}^{(1)} = 1$ for $i \in \{2k + 1, \dots, 5k\}$, $j \in \{3, 4, 5\}$. Now there are at least two blocks of correlated variables in the posterior. For every other variable β_j , $j = 5, \dots, d$, set $X_{ij}^{(1)} = 1$ for $i \in \{jk + 1, \dots, (j + 1)k\}$. In order to enforce dependency between all variables, we perturb every entry of $X^{(1)}$: $X_{ij}^{(1)} = x_{ij}^{(1)} + 0.1\xi_{ij}$, where ξ_{ij} are independent beta distributed variables with parameters $(0.1, 0.1)$.

The second design matrix $X^{(2)}$ is formed as follows. For $i = 1, \dots, n$ and $j = 1, \dots, d$ set

$$X_{ij}^{(2)} = 0.3 \left(\delta_{ij} + \frac{\xi_i}{i} \right),$$

where δ_{ij} is the Kronecker symbol and ξ_i are i.i.d. beta distributed with parameters $(0.1, 0.1)$.

Remark. Correlation matrix of the posterior with the design matrix $X^{(1)}$ has blocks of highly correlated coordinates, whereas in case of the design matrix $X^{(2)}$, the correlation matrix seems to have only moderate non-diagonal entries.

Truncated Multivariate Normal Distribution

Gibbs sampler is a natural algorithm to sample from the TMVN distribution as suggested by Geweke [1991]. We consider linear truncation domain $b_1 \leq Ax \leq b_2$ where $x \in \mathbb{R}^d$, A is some $d \times d$ matrix. Geweke [1991] suggested to transform the underlying normal distribution so that one needs to sample from $N(0, \Sigma_0)$ truncated to a rectangle $c_1 \leq x \leq c_2$.

We set $c_1 = (1, \dots, 1) \in \mathbb{R}^d$, $c_2 = (3, \dots, 3) \in \mathbb{R}^d$ and generate two different covariance matrices Σ_0 .

$$\Sigma_0^{(1)} = \text{Corr} \left(0.01I_d + v_1 v_1^T \right),$$

$$\Sigma_0^{(2)} = \text{Corr} \left(0.01I_d + v_2 v_2^T \right),$$

where $v_1 = (\xi_1, \dots, \xi_d)$, $v_2 = \left(\frac{\xi_1}{\log(2)}, \frac{\xi_2}{\log(3)}, \dots, \frac{\xi_d}{\log(d+1)} \right)$, and ξ_i are i.i.d. beta distributed with parameters $(0.1, 0.2)$. Here $\text{Corr}(M)$ denotes a correlation matrix that corresponds to M .

Note that the truncation domain does not contain the mode of the distribution, making it very different from the non-truncated normal distribution.

Markov Switching Model

Let $x_{1:i} = (x_1, \dots, x_i)$. We consider a version of stochastic volatility model where the underlying chain may be in either high or low volatility mode. Namely, the latent data X_i forms an AR(1) process:

$$X_i \sim N(X_{i-1}, \sigma_{r(i)}^2),$$

where the chain can be in one of the two volatility regimes $r(i) \in \{0, 1\}$. $r(i)$ itself forms a Markov chain with a transition matrix

$$\begin{pmatrix} 1 - a_1 & a_1 \\ a_2 & 1 - a_2 \end{pmatrix}.$$

Here a_1 and a_2 are called switching probabilities and assumed to be known. The observed data Y_i is normally distributed:

$$Y_i \sim N(X_i, \beta^2)$$

with known variance β^2 . We consider data of $n = 100$ observations and aim to sample from the posterior

$$X_1, \dots, X_n, r(1), \dots, r(n) | Y_1, \dots, Y_n.$$

Since $n = 100$, the total number of parameters $d = 200$. We fix $a_1 = 0.001$ and $a_2 = 0.005$.

The underlying hidden Markov chain $(X_i, r(i))$ is obtained as follows. We start chain $r(i)$ at its stationary distribution and $X_0 \sim N(0, \sigma_{r(0)}^2)$. We then randomly generate chain $r(i)$ so that there are two switchings occur. Thus, we obtain $r(i) = 1$ for $i = 57, \dots, 79$, and $r(i) = 0$, otherwise.

We consider data for 3 different combinations of σ_0^2, σ_1^2 , and β^2 :

(a) $\sigma_0^2 = 1, \sigma_1^2 = 10, \beta^2 = 1;$

(b) $\sigma_0^2 = 1, \sigma_1^2 = 10, \beta^2 = 3;$

(c) $\sigma_0^2 = 1, \sigma_1^2 = 5, \beta^2 = 1.$

Full conditionals may be obtained and are easy to sample from. We shall demonstrate performance of the coordinate-wise Gibbs Sampler for this problem.

One might notice that the even and odd blocks of the coordinates can be updated simultaneously, meaning that coordinate-wise updates might be suboptimal. However, we are not motivated to find the best algorithms, but rather to demonstrate that the ARSGS can provide speed up even when the target distribution is discrete. Our intuition is such that the variables around the switching points will mix much slower, meaning that the ARSGS should update those coordinates more frequently.

2.7.1 Adaptive Random Scan Gibbs Sampler

We implement the ARSGS Algorithm 9 for all the examples. To do so, we need to specify a number of parameters in the algorithm. Since in the Euclidean space \mathbb{R}^d distance from the origin to a simplex is $\frac{1}{\sqrt{n}}$, we find it reasonable to choose $a_m = \frac{\log(50\sqrt{d+m})}{50\sqrt{d+m}}$. In fact, one may choose arbitrary positive constant instead of 50. We do not know the right scaling for b_m and thus, set $b_m = a_m$.

We set the lower bound $\varepsilon := \frac{1}{d^2}$. The choice of ε is motivated by Theorem 4, where it is shown that the maximum improvement of the pseudo-spectral gap is d (dimensionality of the space), which can only happen when one of the coordinates gets all of the probability mass. With the above choice of ε , the maximum probability mass that coordinate can get is $1 - \frac{(d-1)}{d^2}$, meaning that we might not be able to identify the optimal selection probabilities. On the other hand, the pseudo-spectral gap that corresponds to the selection probabilities obtained by the adaptive algorithm (with the specified value of $\varepsilon = \frac{1}{d^2}$) will be close to the optimal value.

We choose the sequence k_m to be $k_m = 5000$ for PHM and TMVN examples, and $k_m = 8 \times 10^5$ for MSM. We discuss an effective choice of k_m in Section 2.7.3.

We run the coordinate-wise ARSGS and the vanilla RSGS to obtain 5 million samples with 50 iterations thinning (i.e., we record every 50-th iteration of the chain) in PHM and TMVN examples. For the MSM example thinning is 8000 and number of samples is 10 million in cases (a), (b), and 30 million in the case (c).

Poisson Hierarchical Model

For this example, we show that in a long run the ARSGS not only outperforms the vanilla RSGS but performs similarly to the RSGS with the pseudo-optimal weights (that are estimated from the adaptive chain run).

The data Y_i is generated separately for the design matrices $X^{(1)}$ and $X^{(2)}$.

We summarise results in Tables 2.1 and 2.2, respectively. In a view of the von Mises theorem (see van der Vaart [2000]), we expect the ARSGS to work well in both examples. One can observe reduction in the maximum asymptotic variance over the linear functions (2.36) by 9.27 and 6.97 times respectively.

In the example with the design matrix $X^{(1)}$, the correlation between the 1st and 2nd coordinate is -0.98 and they are both nearly uncorrelated with the other coordinates. However, the corresponding optimal weights are such that $p_1^{\text{opt}} \approx p_2^{\text{opt}} \approx 0.085$. The most of the probability mass is put on the coordinate 5: $p_5^{\text{opt}} \approx 0.29$. The maximum correlation of the coordinate 5 with other directions is at most 0.57 in absolute value. In fact, excluding coordinates 1, 2 and 5, all the off-diagonal correlations do not exceed 0.57 in absolute value.

For the second example with the design matrix $X^{(2)}$, all the correlations are less than 0.21 . In some sense this example is consistent with the toy Example 2 from Section 2.3. Here $p_1^{\text{opt}} \approx 0.188$, whereas all other optimal selection probabilities are in a range between 0.001 and 0.01 .

Note that the RSGS with the optimal weights performs nearly the same as the adaptive counterpart. For each design matrix ACF plots are produced for two coordinates with high and low optimal weights in Figures 2.1a and 2.1b.

Empirically, we observe that the adaptive algorithm tries to allocate the selection probabilities in such a way that the effective number of independent samples (effective sample size) for every direction is the same. Hence, all the coordinates have about the same autocorrelation function (see Figure 2.1 and 2.2). Note that the autocorrelation changes proportionally to the reduction in the asymptotic variance.

Table 2.1: PHM. Gibbs (d=50). Example 1

	1/ P-Gap	$\max_i \sigma_{as}^2(l_i)$
vanilla	13435	482
adaptive	1355	52
optimal	-	54
$\frac{\text{vanilla}}{\text{adaptive}}$	9.9	9.27

Table 2.2: PHM. Gibbs (d=50). Example 2

	1/ P-Gap	$\max_i \sigma_{as}^2(l_i)$
vanilla	7340	272
adaptive	919	39
optimal	-	40
$\frac{\text{vanilla}}{\text{adaptive}}$	7.97	6.97

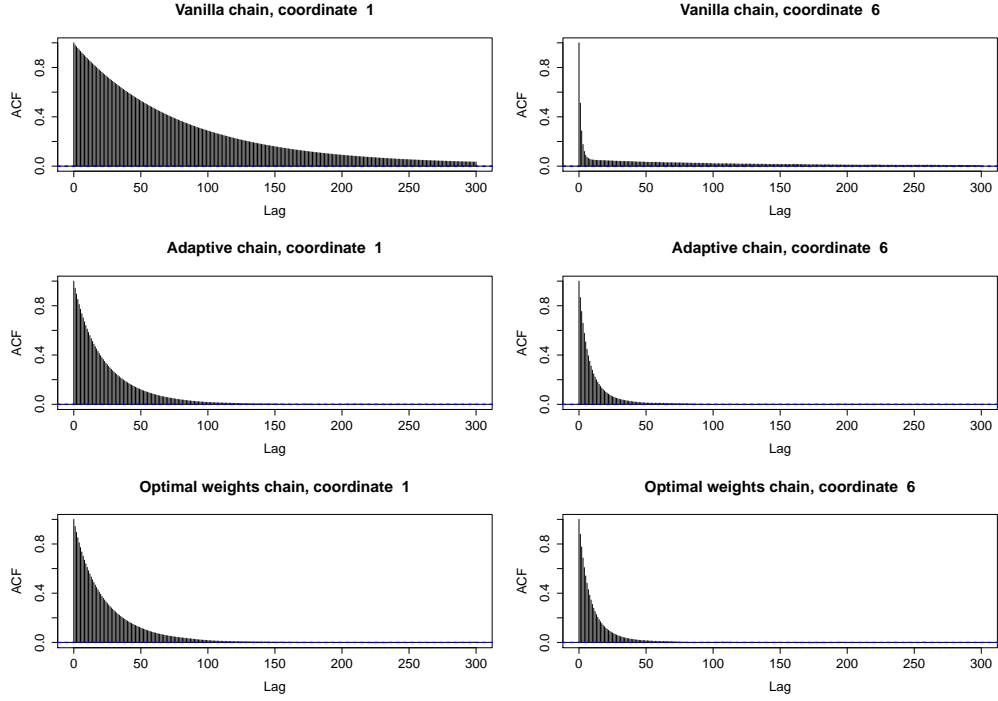
Remark. We use the Adaptive Rejection Sampling algorithm (see [Gilks and Wild \[1992\]](#)) in order to sample from the full conditionals. Since the normalising constant is not known in this case, we could not establish geometric ergodicity of the RSGS in this case. On the other hand, results of Latuszynski et al. [Latuszyński et al. \[2013b\]](#) ensure that the RWMwG is geometrically ergodic. Since typically the RSGS converges faster than the corresponding RWMwG, we suggest that the RSGS is also geometrically ergodic for the Poisson Hierarchical Model. This means that heuristically the modified ARGs Algorithm [14](#) is ergodic in the current settings.

Truncated Multivariate Normal Distribution

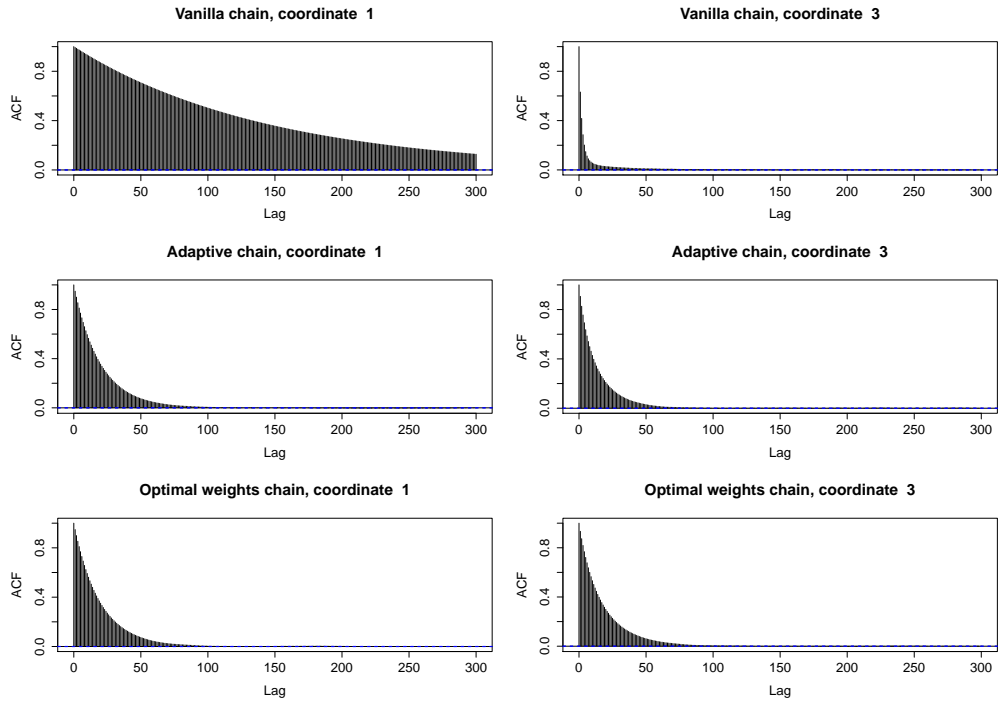
For the first correlation matrix $\Sigma_0^{(1)}$ the reduction in the maximum asymptotic variance over the linear functions([2.36](#)) is 3.44, which is surprisingly very close to the ratio of the pseudo-spectral gaps (Table [2.3](#)). The autocorrelation plot of 2nd and 47th coordinates is in Figure [2.2](#). The same effect of keeping the same autocorrelations for all the coordinates is observed.

Table 2.3: TMVN ($d = 50$). Example 1

	1/ P-Gap	$\max_i \sigma_{as}^2(l_i)$
vanilla	6449	239
adaptive	1857	72
$\frac{\text{vanilla}}{\text{adaptive}}$	3.47	3.32



(a) Example 1



(b) Example 2

Figure 2.1: $d = 50$. PHM. ACF

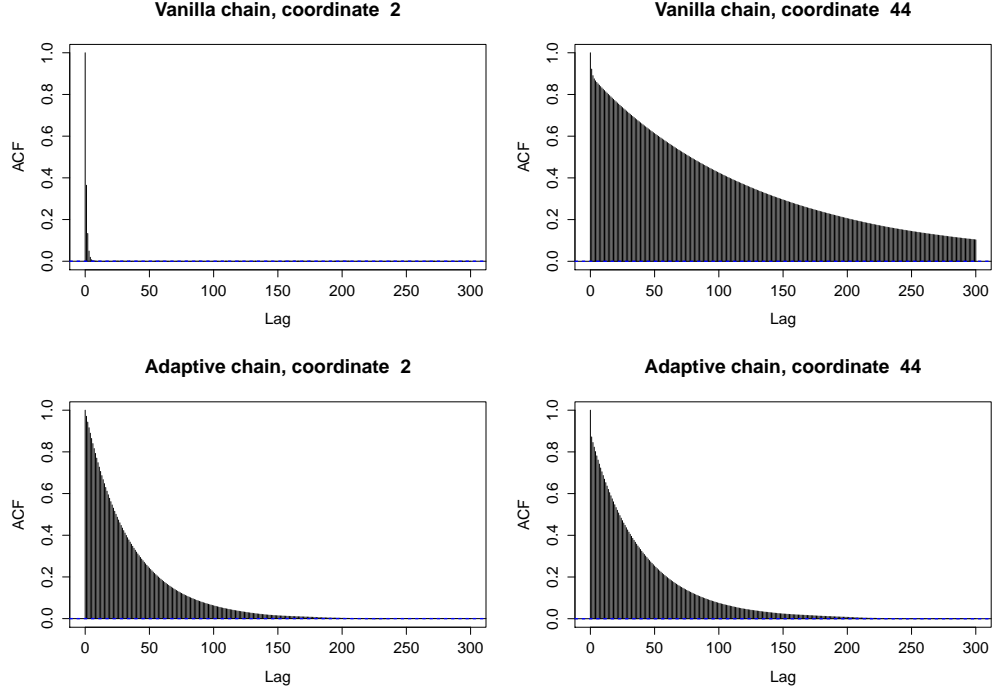


Figure 2.2: $d = 50$. TMVN. ACF. Example 1

For the second correlation matrix $\Sigma_0^{(2)}$, the improvement of the asymptotic variance is only half of the improvement of the spectral gap, as seen from Table 2.4. However, we still observe that the ARSGS assigns more weight to the coordinates that mix slower.

Table 2.4: TMVN ($d = 50$). Example 2

	1/P-Gap	$\max_i \sigma_{as}^2(l_i)$
vanilla	467	12.6
adaptive	161	8.3
$\frac{\text{vanilla}}{\text{adaptive}}$	2.9	1.5

Remark. In this example, the RSGS kernels (2.1) satisfy the uniform minorisation condition (A0). The corresponding small set is the whole domain, because it is compact. The diminishing adaptation condition (C1) holds by con-

struction. Therefore, the ARSGS algorithm is ergodic by virtue of Theorem 4.2 [Latuszyński et al. \[2013b\]](#).

Markov Switching Model

Note that half of the coordinates in the target distribution are discrete. The naive estimator (2.16) of the covariance structure is often singular (i.e., non-invertible) in these settings. Therefore, to implement the ARSGS, in Step 2 of the Algorithm 9, we use a perturbed naive estimator $\widehat{\Sigma}_n + \frac{1}{d^3}I_d$, where $d = 200$ is dimensionality of the target distribution. We did not observe any significant impact of the added perturbation on the estimated optimal selection probabilities.

We expect that if the dependency structure is described by the correlations (i.e., zero correlation implies weak in some sense dependency), then the ARSGS shall outperform the vanilla RSGS. We observe that this happens in cases (a) and (b). More precisely, the ARSGS tends to put more weight on coordinates that have larger asymptotic variance and results are found in Table 2.5, where the improvement of the pseudo spectral gap for each case is presented, and in Table 2.6, where the corresponding improvement in asymptotic variance over the linear functions (2.36) is presented. Notice, in all cases the maximum asymptotic variance $\max_i \sigma_{as}^2(l_i)$ is attained for the coordinate i that corresponds to some discrete direction $r(i)$.

Table 2.5: MSM ($d = 200$). 1/P-Gap

	(a)	(b)	(c)
vanilla	18875	71106	117127
adaptive	3450	9924	19301
$\frac{\text{vanilla}}{\text{adaptive}}$	5.47	7.17	6.07

Table 2.6: MSM ($d = 200$). $\max_i \sigma_{as}^2(l_i)$

	(a)	(b)	(c)
vanilla	21.7	45.7	197
adaptive	6	12.6	204
$\frac{\text{vanilla}}{\text{adaptive}}$	3.6	3.63	0.97

The case (c) is special in a sense that the correlation structure does not reveal the dependency structure. Here the maximum selection probability is assigned to the coordinate that corresponds to the variable $r(99)$. The asymptotic variance for the corresponding linear function l_i drops roughly by 4.5 times from 40.56 to 9.06. However, the maximum asymptotic variance over the linear functions (2.36) is attained on the coordinate i that corresponds to $r(64)$. The estimated optimal weight p_i^{opt} corresponding to $r(64)$ is only slightly larger than the uniform weight $1/200$.

To justify convergence of the ARSGS and the results in Tables 2.5 and 2.6, we provide a proof of the geometric ergodicity.

Proposition 4. *In cases (a), (b) and (c), the RSGS for the Markov Switching Model described above is geometrically ergodic, i.e., satisfies (A2).*

2.7.2 Adaptive Random Walk Metropolis within Adaptive Gibbs

As before, we sample from the same PHM in \mathbb{R}^{50} . We consider the same algorithm settings for the ARWMwAG Algorithm 12 as for the ARSGS Algorithm 9, and also set parameter $q := 1$. We compare performance of the Random Walk Metropolis-within-Gibbs algorithm with its adaptive versions ARWMwG and ARWMwAG. For the RWMwG, the proposal variances β_i are chosen to be ones.

Poisson Hierarchical Model

Tables 2.7 and 2.8 provide the analysis of the asymptotic variances. For the first design matrix $X^{(1)}$ we observe a 7 times improvement of the ARWMwAG over the ARWMwG algorithm, and the total improvement of almost 15 times over the non-adaptive RWMwG. For the second design matrix $X^{(2)}$, the corresponding improvement is 6.1 and 12.3 times respectively. On Figure 2.3 we present the improvements to the ACF.

Table 2.7: PHM. MwG ($d = 50$). Example 1

	1/ P-Gap	$\max_i \sigma_{as}^2(l_i)$
RWMwG (non-adaptive)	–	1993
ARWMwG (partially adaptive)	13244	971
ARWMwAG (fully adaptive)	1376	138
$\frac{\text{partially adaptive}}{\text{fully adaptive}}$	9.63	7
$\frac{\text{non-adaptive}}{\text{fully adaptive}}$	–	14.45

Table 2.8: PHM. MwG ($d = 50$). Example 2

	1/ P-Gap	$\max_i \sigma_{as}^2(l_i)$
RWMwG (non-adaptive)	–	1276
ARWMwG (partially adaptive)	7461	639
ARWMwAG (fully adaptive)	970	104
$\frac{\text{fully adaptive}}{\text{partially adaptive}}$	7.69	6.14
$\frac{\text{fully adaptive}}{\text{non-adaptive}}$	–	12.27

Remark. The target distribution satisfies the Assumption 5.4 of [Latuszyński et al. \[2013b\]](#). If one chooses the proposal in Step 2 of RWMwG Algorithm 10 to be a mixture of normals, i.e., $0 < q < 1$, or restricts the proposal variances β_i to be in some interval $[c_1, c_2]$, $\infty > c_2 > c_1 > 0$, then the ARWMwG and ARWMwAG is ergodic by the virtue of Theorem 5.5 of [Latuszyński et al. \[2013b\]](#).

2.7.3 Computational cost of the adaptation

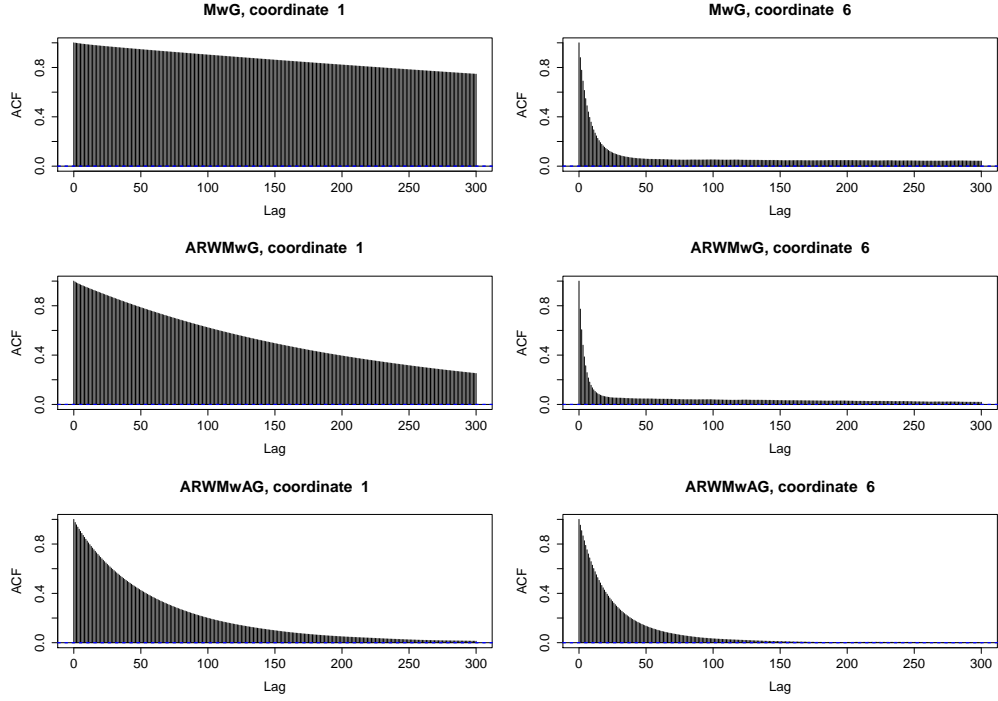
By doing the adaptations of an MCMC algorithm we increase the total running time of the algorithm. Complexity of the projection Algorithm 8 is bounded by the complexity of a sorting algorithm used in Step 4, which is usually of order $\mathcal{O}(d \log(d))$. Thus, one can easily see that the total adaptation cost of Steps 2 - 4 of the ARSGS Algorithm 9 is bounded by the complexity of the Step 2, which requires finding diagonal blocks of the inverted covariance matrix. Usual Gauss matrix inversion is of order $\mathcal{O}(d^3)$. In high dimensional settings it is an expensive procedure. However, one can choose the sequence k_m in the setting of the ARSGS Algorithm 9 in order to make the adaptation cost negligible comparing to the sampling Step 1.

Turn back to the Poisson Hierarchical Model example with the design matrix $X^{(1)}$. The sequence k_m was chosen to be $k_m = 5000$. In column 2 of Table 2.9 we put the average real time in seconds spent on sampling Step 1 of the ARSGS and ARWMwAG algorithms. The average time spent for one adaptation (i.e., to perform Steps 2 - 4 of the ARSGS Algorithm 9) is in column 3. The maximum asymptotic variance over the linear functions (2.36) is in column 1.

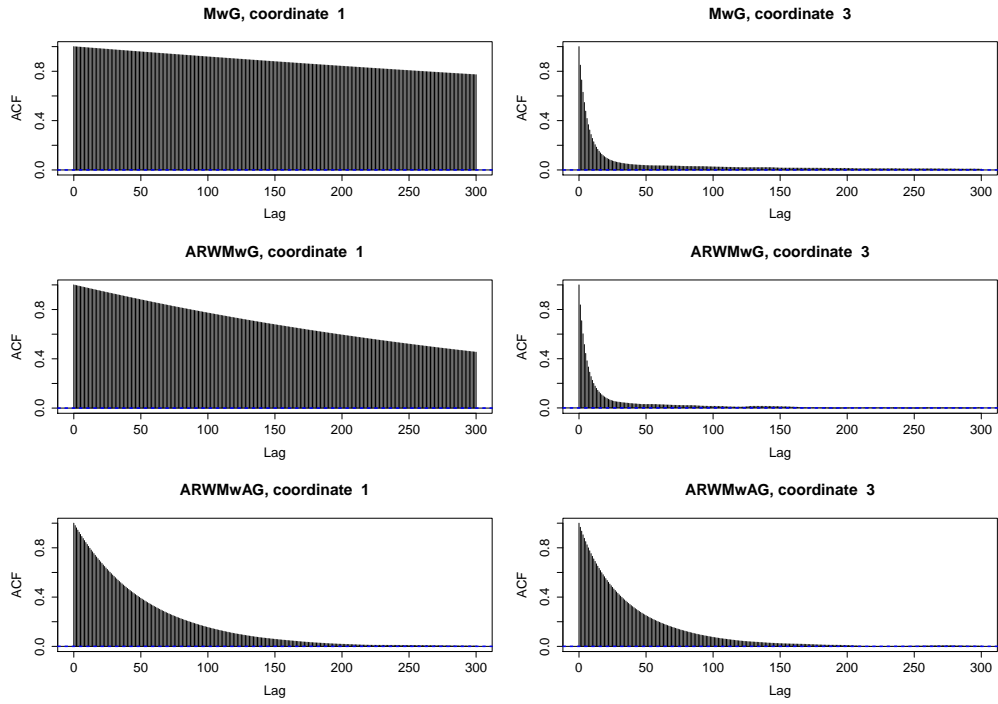
Table 2.9: PHM. Example 1 ($d = 50$)

	$\max_i \sigma_{as}^2(l_i)$	Cost per 5000 iterations	Cost of adaptation
ARSGS	52	0.37	0.0025
ARWMwAG	138	0.028	0.0025

Gibbs Sampling for the PHM requires the use of the adaptive rejection sampling (see [Doss and Narasimhan \[1994\]](#), [Gilks and Wild \[1992\]](#)), which significantly increases the time needed to obtain a sample. Therefore, even though the ARSGS has 2.65 times lower asymptotic variance than the ARWMwAG algorithm, it samples more than 10 times slower.



(a) Example 1



(b) Example 2

Figure 2.3: $d = 50$. PHM. ACF

By adjusting the sequence k_m one can tune the ratio of the adaptation time over the sampling time. In fact, the sampling and adaptations can be performed independently in a sense that they may be computed on different CPUs as demonstrated in the Algorithm 15.

Algorithm 15: Parallel versions of ARSGS and ARWMwAG

Set all initial parameters for the ARSGS (ARWMwAG).

Do on different CPUs:

- $n = n + k_i$. $p^i = \frac{w^i}{w_1^i + \dots + w_s^i}$ Run RSGS(p^i) (or ARWMwG(p^i)) for k_i steps.
- Do steps the steps 2 - 4 of ARSGS based on available chain output.

Wait till both CPUs finish their jobs. Then iterate the procedure.

2.8 Discussion

We have devised the Adaptive Random Scan Gibbs and Adaptive Random Walk Metropolis within Adaptive Gibbs algorithms, where adaptations are guided by optimising the L_2 -spectral gap for the Gaussian target analogue called pseudo-spectral gap. The performance of the adaptive algorithms has been studied in Section 2.7. We have seen that it might hard to decide in advance whether the adaptive algorithm would outperform the non-adaptive counterpart. On the other hand, as suggested in Section 2.7.3, the computational time added by the adaptation can be made negligible comparing to the total run time of the algorithm. Therefore, we believe that it is reasonable to utilise the adaptive algorithms given that substantial computational gain may be achieved. However, one needs a natural notion of the covariance structure for the target distribution in order to implement the adaptive algorithms.

We have analysed ergodicity property of the adaptive algorithms in Section 2.6. We have developed a concept of the local simultaneous drift condition (A3). We have shown in Theorem 5 that the condition is natural for the ARSGS. Under this condition, in Theorem 7 we have established ergodicity of modified AMCMC Algorithms 13. In particular, in Theorem 8 we have proved ergodicity of the modified ARSGS Algorithm 14.

In order to establish convergence in Theorem 8, we do not require the sequence of estimated optimal sampling probabilities weights p^n to converge at all. Instead, we require only the diminishing condition to hold, i.e., $|p^n - p^{n-1}| \xrightarrow{P} 0$

as $n \rightarrow \infty$. In fact, it is not clear whether the estimated weights converge to the pseudo-optimal ones, even if one knows the target covariance matrix. Empirically, for numerous examples, we have observed that the adapted selection probabilities do converge to a unique solution, where the uniqueness is guaranteed by Theorem 3.

Open problem. Assume that the covariance matrix of the target distribution is known, i.e., $\hat{\Sigma}_n = \Sigma$ for all n . Prove that the estimated weights p^i in the ARSGS algorithm converge to the pseudo-optimal weights p^{opt} .

We emphasise that there is no universal algorithm to optimise the pseudo-spectral gap function (2.18), given that the covariance structure Σ is unknown.

Various other modification of the ARSGS algorithm are possible. For instance, we can think of using some other optimisation algorithm instead of the subgradient method (described in Algorithm 6) in order to estimate the pseudo-optimal weights (2.10). Also, the user may know the structure of the covariance matrix Σ in advance, so that the naive estimator (2.16) could be improved. For example, if the covariance matrix is banded, a more efficient threshold estimator should be used (see Bickel and Levina [2008]).

In Section 3.4.1 of Chapter 3 we discuss a modified (Air) version of the ARSGS Algorithm 14, for which we prove the SLLN and the MSE convergence under the local simultaneous geometric drift assumption (A3). If, additionally, the sequence of adapted selection probabilities p^n converges, we derive the CLT.

2.9 Proofs of the statements from Chapter 2

Proof of Lemma 1. The proof is a modification of Theorem 1 of Amit [1996], and thus, we outline only the key points.

One can easily check that $\{H_\alpha(Kx)\}$ form an orthonormal system in $L_2(\mathbb{R}^d, \pi)$ using the definition. From Theorem 6.5.3 of Andrews et al. [1999], it follows that one dimensional Hermite polynomials h_k form a complete orthogonal basis of $L_2(\mathbb{R}, \exp(-x^2/2))$, implying $\{H_\alpha(Kx)\}$ form an orthogonal basis of $L_2(\mathbb{R}^d, \pi)$.

For $c \in \mathbb{R}^d$, define a generating function

$$f_c(x) := \sum_{\alpha} c^{\alpha} \frac{H_{\alpha}(Kx)}{\sqrt{\alpha!}}, \quad (2.39)$$

where $c^{\alpha} := c_1^{\alpha_1}, \dots, c_d^{\alpha_d}$ and $0^0 := 1$.

From Section 4.2.1 of Roman [1984], we know that the generating function

can be represented as

$$f_c(x) = \exp \left(\langle c, Kx \rangle - \frac{\|c\|^2}{2} \right).$$

Recall that Pr_i stands for full conditional update of $x_i = (x_{i1}, \dots, x_{ir_i})$ from its full conditional. For functions $f \in L_2(\mathbb{R}^d, \pi)$, let

$$(Pr_i f)(x) := \int f(y_i, x_{-i}) \pi(y_i | x_{-i}) dy_i.$$

Define $T^{(i)} := I - KD_i K$. Note that $T^{(i)} = \left(T_{ij}^{(i)} \right)_{i,j=1}^d$ is a d -dimensional matrix.

The key property to prove the first part of the lemma is the following statement that can be obtained via direct calculations.

Lemma 3. *For all $c \in \mathbb{R}^d$*

$$(Pr_i f_c)(x) = f_{T^{(i)}c}(x),$$

where f_c is defined in (2.39).

Let Π_k be the set of all sequences $(\varepsilon_1, \dots, \varepsilon_k)$ of length k with elements from $\{1, \dots, d\}$. Partition Π_k into equivalence classes R_α , $\alpha = (\alpha_1, \dots, \alpha_d) \in Z_+^d$, $|\alpha| = k$, such that $\varepsilon = (\varepsilon_1, \dots, \varepsilon_k) \in R_\alpha$ if and only if the sequence ε has α_1 1's, \dots , α_d d's. In other words, R_α forms a set of all permutations of the elements of $(\varepsilon_1, \dots, \varepsilon_k)$, implying that the number of elements in R_α is $|R_\alpha| = \frac{k!}{\alpha!}$.

Lemma 3 implies

$$Pr_i \left(\sum_{\alpha} c^{\alpha} \frac{H_{\alpha}(K \cdot)}{\sqrt{\alpha!}} \right) (x) = \sum_{\alpha} (T^{(i)}c)^{\alpha} \frac{H_{\alpha}(Kx)}{\sqrt{\alpha!}}. \quad (2.40)$$

Fix β such that $|\beta| = k$. For each α , $|\alpha| = k$, fix some representative $\sigma = \sigma(\alpha) \in R_\alpha$. Rewrite $T^{(i)}c$ as

$$T^{(i)}c = \left(T_{j1}^{(i)}c_1 + \dots + T_{jd}^{(i)}c_d \right)_{j=1}^d.$$

Since $c \in \mathbb{R}^d$ is arbitrary, the coefficient of c^{β} on both sides of (2.40) should coincide for all $\beta \in Z_+^d$, providing a formula for the image of $H_{\beta}(Kx)$:

$$Pr_i \left(\frac{H_\beta(K \cdot)}{\sqrt{\beta!}} \right) (x) = \sum_{|\alpha|=k} \frac{1}{\sqrt{\alpha!}} \left(\sum_{\varepsilon \in R_\beta} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right) H_\alpha(Kx).$$

Since $\sigma = \sigma(\alpha)$ was chosen arbitrary, the above sum is equal to

$$\begin{aligned} Pr_i \left(\frac{H_\beta(K \cdot)}{\sqrt{\beta!}} \right) (x) &= \sum_{|\alpha|=k} \frac{1}{|R_\alpha| \sqrt{\alpha!}} \left(\sum_{\varepsilon \in R_\beta, \sigma \in R_\alpha} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right) H_\alpha(Kx) \\ &= \sum_{|\alpha|=k} \frac{\sqrt{\alpha!}}{k!} \left(\sum_{\varepsilon \in R_\beta, \sigma \in R_\alpha} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right) H_\alpha(Kx). \end{aligned}$$

We conclude that

$$Pr_i(H_\beta(K \cdot))(x) = \sum_{|\alpha|=k} \frac{\sqrt{\alpha!} \sqrt{\beta!}}{k!} \left(\sum_{\varepsilon \in R_\beta, \sigma \in R_\alpha} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right) H_\alpha(Kx), \quad (2.41)$$

implying the first part of the Lemma.

We are left to show that the maximum eigenvalue of P_p on S_k is non-increasing, revealing that the second largest eigenvalue of P_p is attained on S_1 .

Recall that $\{R_\alpha \mid \alpha \in Z_+^n, |\alpha| = k\}$ form a partition of all possible sequences $(\varepsilon_1, \dots, \varepsilon_k)$, $\varepsilon_i \in \{1, \dots, d\}$. Moreover, as we have just seen, Pr_i is invariant on S_k (S_k is defined in the statement of the lemma), that is Pr_i acts like a matrix on S_k . (2.41) implies that Pr_i can be represented as

$$Pr_i(H_\beta(K \cdot))(x) = \sum_{|\alpha|=k} \frac{1}{\sqrt{|R_\alpha| |R_\beta|}} \left(\sum_{\varepsilon \in R_\beta, \sigma \in R_\alpha} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right) H_\alpha(Kx),$$

implying that the matrix that corresponds to Pr_i consists of entries

$$\frac{1}{\sqrt{|R_\alpha| |R_\beta|}} \left(\sum_{\varepsilon \in R_\beta, \sigma \in R_\alpha} T_{\sigma_1 \varepsilon_1}^{(i)} \cdots T_{\sigma_k \varepsilon_k}^{(i)} \right)$$

Thus, we have shown that on S_k , P_p acts as a matrix with corresponding entries

obtained as normalised block sums of

$$F_k = \sum_{i=1}^s p_i (T^{(i)})^{\otimes k},$$

where $(T^{(i)})^{\otimes k}$ is the k -th Kronecker product of $T^{(i)}$ (i.e., the k -th tensor product, see [Reed and Simon \[1980\]](#), VIII.10).

The next statement is Lemma 1 from [Amit \[1996\]](#) and we do not prove it.

Lemma 4. *Let A be a non-negative definite $r \times r$ matrix. Let R_1, \dots, R_q be a partition of $\{1, \dots, r\}$. Define matrix B to be the $q \times q$ matrix,*

$$B_{kl} = \frac{1}{\sqrt{|R_k||R_l|}} \sum_{i \in R_k, j \in R_l} A_{ij}.$$

Then the maximum eigenvalue of B is less or equal than the maximum eigenvalue of A .

Lemma 4 shows that the maximum eigenvalue of P_p restricted to S_k is dominated by the maximum eigenvalue of F_k .

Rewrite F_{k+1} as a difference of two positive semi-definite operators

$$F_{k+1} = F_k \otimes I - \sum_{i=1}^s p_i (T^{(i)})^{\otimes k} \otimes (I - T^{(i)})$$

It follows that for all $k \geq 0$, $F_k \otimes I \geq F_{k+1}$ (i.e., for all vectors x , $\langle F_k \otimes I x, x \rangle \geq \langle F_{k+1} x, x \rangle$). Since $\|F_k\| = \|F_k \otimes I\|$ (see [Reed and Simon \[1980\]](#), VIII.10), the largest eigenvalue of F_k (that is equal to largest eigenvalue of $F_k \otimes I$) is greater than the one of the operator F_{k+1} . Thus, the largest eigenvalues of $P_p|_{S_k}$, $k \geq 0$ form a non-increasing sequence.

Note that $P_p|_{S_0}$ corresponds to the unit eigenvalue and that the matrix that corresponds to $P_p|_{S_1}$ is exactly F_1 , as easily seen from (2.41). Therefore, the second largest eigenvalue of P_p is attained on S_1 and is equal to the maximum eigenvalue of F_1 .

□

Proof of Theorem 1. From the formula (2.41) it follows that for $k = 1$, a matrix that corresponds to $P_p|_{S_1}$ is exactly F_1 .

□

Proof of Lemma 2. Let $\lambda \neq 0$ such that

$$ABx = \lambda x$$

for some non-zero x . Multiply both sides by B

$$BA(Bx) = \lambda Bx.$$

If $\lambda = 0$ and $x \neq 0$ such that

$$ABx = 0,$$

we may have either $Bx \neq 0$ or $Bx = 0$. In the first case multiply both sides by B so that $BA(Bx) = 0$. Otherwise, if A is invertible, find y such that $Ay = x$ so that $BAy = 0$. If A is not invertible, there exists $y \neq 0$ such that $Ay = 0$ so that again $BAy = 0$.

□

Proof of Theorem 3. Define

$$h(p) = \lambda_{\min}(D_p Q) = \lambda_{\min}(\sqrt{Q} D_p \sqrt{Q}). \quad (2.42)$$

Assume p_1^{opt} and p_2^{opt} are two different points that maximise h . Then a function

$$g(\alpha) := h(\alpha p_1^{\text{opt}} + (1 - \alpha)p_2^{\text{opt}})$$

is constant on $[0, 1]$ due to concavity of h (see Proposition 3) and equals, say, λ . Since $h(p)$ is itself the minimum eigenvalue of $\sqrt{Q} D_p \sqrt{Q}$, there exist unit vectors x_0, x_1, x_2 such that

$$\begin{aligned} \sqrt{Q} D_{\frac{1}{2}p_1^{\text{opt}} + \frac{1}{2}p_2^{\text{opt}}} \sqrt{Q} x_0 &= \lambda x_0 \\ \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_1 &= \lambda x_1 \\ \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_2 &= \lambda x_2 \end{aligned}$$

Since g is constant on $[0, 1]$,

$$\begin{aligned} \frac{1}{2} \langle \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_0, x_0 \rangle + \frac{1}{2} \langle \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_0, x_0 \rangle &= g\left(\frac{1}{2}\right) = \frac{g(0)}{2} + \frac{g(1)}{2} \\ &= \frac{1}{2} \langle \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_1, x_1 \rangle + \frac{1}{2} \langle \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_2, x_2 \rangle \end{aligned}$$

$$\leq \frac{1}{2} \left\langle \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_0, x_0 \right\rangle + \frac{1}{2} \left\langle \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_0, x_0 \right\rangle,$$

where the last inequality holds since x_1, x_2 are the minimum eigenvectors. Hence,

$$\begin{aligned} \left\langle \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_0, x_0 \right\rangle &= \left\langle \sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_1, x_1 \right\rangle = \lambda, \\ \left\langle \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_0, x_0 \right\rangle &= \left\langle \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_2, x_2 \right\rangle = \lambda. \end{aligned}$$

Therefore,

$$\sqrt{Q} D_{p_1^{\text{opt}}} \sqrt{Q} x_0 = \lambda x_0 \text{ and } \sqrt{Q} D_{p_2^{\text{opt}}} \sqrt{Q} x_0 = \lambda x_0, \quad (2.43)$$

which follows from the following simple statement.

Lemma 5. *Let A be a $n \times n$ symmetric matrix, and x be a unit vector, such that $\langle Ax, x \rangle = \lambda_{\min}(A)$. Then $Ax = \lambda_{\min}(A)x$.*

Let $y_0 = \sqrt{Q} x_0$. Then (2.43) is equivalent to

$$D_{p_1^{\text{opt}}} y_0 = \lambda \Sigma y_0 \text{ and } D_{p_2^{\text{opt}}} y_0 = \lambda \Sigma y_0.$$

Using the definition of (2.7), the last equalities yield

$$(p_i^{\text{opt}})_j = \lambda \frac{\left\langle (\Sigma y_0)_j, (y_0)_j \right\rangle}{\left\langle Q_{jj}^{-1} (y_0)_j, (y_0)_j \right\rangle},$$

for $i = 1, 2$, if $(y_0)_j \neq 0$.

Let $p := \frac{1}{2} p_1^{\text{opt}} + \frac{1}{2} p_2^{\text{opt}}$. It is left to show that for every $j \in \{1, \dots, s\}$, one can find a minimum eigenvector x_0 of $\sqrt{Q} D_p \sqrt{Q}$, such that for the corresponding vector $y_0 = \sqrt{Q} x_0$, we have $(y_0)_j \neq 0$.

Define a space $S_0 = \{x_0 | \sqrt{Q} D_p \sqrt{Q} x_0 = \lambda x_0\}$ as a space generated by all the minimum eigenvectors of $\sqrt{Q} D_p \sqrt{Q}$.

Assume, on the contrary, that for some $j \in \{1, \dots, s\}$, and all $x_0 \in S_0$, we have $(y_0)_j = 0$.

Define a space $S_{\perp} := \{x | x \text{ is orthogonal to } S_0\}$. For $\varepsilon \geq 0$, let

$$A_{\varepsilon} := \sqrt{Q} D_{p^{(\varepsilon)}} \sqrt{Q},$$

where $p_k^{(\varepsilon)} = (1 + \varepsilon) p_k$ if $k \neq j$, and $p_j^{(\varepsilon)} = p_j - \varepsilon \sum_{k \neq j} p_k$. Note that for all $x_0 \in S_0$

and $\varepsilon \geq 0$,

$$A_\varepsilon x_0 = \sqrt{Q} D_{p^{(\varepsilon)}} \sqrt{Q} x_0 = \sqrt{Q} D_{p^{(\varepsilon)}} y_0 = (1 + \varepsilon) \lambda \sqrt{\Sigma} y_0 = (1 + \varepsilon) \lambda x_0, \quad (2.44)$$

since $y_0 = \sqrt{Q} x_0$, $D_p y_0 = \lambda \Sigma y_0$, and $(y_0)_j = 0$ by the assumption. That is, $(1 + \varepsilon) \lambda$ is an eigenvalue of A_ε , and S_0 is the subspace of the corresponding eigenvectors.

Also, S_\perp is invariant under A_ε (i.e., $A_\varepsilon S_\perp \subset S_\perp$). Indeed, for all $x \in S_\perp$ and $x_0 \in S_0$,

$$\langle A_\varepsilon x, x_0 \rangle = \langle x, A_\varepsilon x_0 \rangle = \lambda(1 + \varepsilon) \langle x, x_0 \rangle = 0.$$

Let $\lambda^{(\varepsilon)}$ be the minimum eigenvalue of A_ε restricted to the space S_\perp . Since S_0 contains all possible minimum eigenvectors of $\sqrt{Q} D_p \sqrt{Q}$, we have $\lambda^{(0)} > \lambda$. Therefore, we can find small enough $r > 0$, such that

$$\lambda^{(0)} > (1 + r) \lambda.$$

Recall that $\lambda^{(\varepsilon)}$ is a continuous function of ε (since it is a concave function by the Proposition 3). Thus, there exists $\delta \in (0, r)$, such that for all $\varepsilon \in [0, \delta]$, $\lambda^{(\varepsilon)} > (1 + r) \lambda$ and also $p^{(\varepsilon)} \in \Delta_{s-1}$. In particular,

$$\lambda^{(\delta)} > (1 + r) \lambda > (1 + \delta) \lambda.$$

Since $S_\perp \cup S_0 = \mathbb{R}^d$, and A_δ is a symmetric, invariant operator on S_\perp and S_0 , we obtain,

$$\lambda_{\min}(A_\delta) = \min\{\lambda_{\min}(A_\delta|_{S_\perp}), \lambda_{\min}(A_\delta|_{S_0})\} = \min\{\lambda^{(\delta)}, (1 + \delta) \lambda\} = (1 + \delta) \lambda,$$

meaning $\lambda(1 + \delta)$ is the minimum eigenvalue of $A_\delta = \sqrt{Q} D_{p^{(\delta)}} \sqrt{Q}$, $p^{(\delta)} \in \Delta_{s-1}$. Hence λ is not the maximum of (2.42), which contradicts to the definition of λ . Thus, there exists $x_0 \in S_0$, such that $(y_0)_j \neq 0$.

□

Proof of Lemma 5. Let $\{x_i\}$ be an orthonormal basis of eigenvectors of A with $\{\lambda_i\}$ being the corresponding eigenvalues. Then $x = \sum_{i=1}^n \langle x, x_i \rangle x_i$. We need to show that $\langle x, x_i \rangle = 0$ for all x_i that are not the minimum eigenvectors. Assume there are at least two vectors x_k and x_j such that $\lambda_k \neq \lambda_j$, $\langle x, x_k \rangle^2 > 0$, and $\langle x, x_j \rangle^2 > 0$.

Then

$$\langle Ax, x \rangle = \sum_{i=1}^n \lambda_i \langle x, x_i \rangle^2 > \lambda_{\min}(A) \sum_{i=1}^n \langle x, x_i \rangle^2 = \lambda_{\min}(A),$$

contradicting the assumption that $\langle Ax, x \rangle = \lambda_{\min}(A)$.

□

Proof of Theorem 4. Let P_p be the Gibbs kernel as in (2.1) with corresponding weights p and let $P_{\frac{1}{s}}$ be the kernel of the vanilla chain. For functions $f, g \in L_2(\mathbb{R}^d, \pi)$ let

$$\langle f, g \rangle = \int f g d\pi, \quad \|f\|^2 = \int f^2 d\pi.$$

Using an equivalent representation for the spectral gap (see a remark to Theorem 2 of [Roberts and Rosenthal \[1997\]](#)), inequality (2.11) is equivalent to

$$\inf_{\|f\|=1, \pi(f)=0} \left\langle (I - P_p)f, f \right\rangle \leq \max_i \left(\frac{p_i}{q_i} \right) \inf_{\|f\|=1, \pi(f)=0} \left\langle (I - P_q)f, f \right\rangle$$

It suffices to establish

$$\left\langle (I - P_p)f, f \right\rangle \leq \max_i \left(\frac{p_i}{q_i} \right) \left\langle (I - P_q)f, f \right\rangle$$

for all f , $\|f\| = 1, \pi(f) = 0$. Let $j = \operatorname{argmax}_i \frac{p_i}{q_i}$. Using the representation (2.1) of P_p , the last inequality is equivalent to

$$\sum_{i=1}^s p_i \left(\frac{q_i}{p_i} - \frac{q_j}{p_j} \right) \langle P_{r_i} f, f \rangle + \frac{q_j}{p_j} \|f\|^2 \leq \|f\|^2.$$

Since $\frac{q_j}{p_j} \leq \frac{q_i}{p_i}$ and $\langle P_{r_i} f, f \rangle \leq \|f\|^2$, the last inequality follows:

$$\sum_{i=1}^s p_i \left(\frac{q_i}{p_i} - \frac{q_j}{p_j} \right) \langle P_{r_i} f, f \rangle + \frac{q_j}{p_j} \|f\|^2 \leq \sum_{i=1}^s p_i \left(\frac{q_i}{p_i} - \frac{q_j}{p_j} \right) \|f\|^2 + \frac{q_j}{p_j} \|f\|^2 = \|f\|^2,$$

where in the last equality we used the fact that $\sum_{i=1}^s p_i = \sum_{i=1}^s q_i = 1$.

□

Proof of Proposition 1. For $i = 1, \dots, k$ let

$$A_i = \begin{pmatrix} p_{2i-1} & p_{2i-1}\rho_i \\ p_{2i}\rho_i & p_{2i} \end{pmatrix}.$$

One can see that the pseudo-optimal weights p^{opt} satisfy

$$p^{\text{opt}} = \operatorname{argmax}_{p \in \Delta_{2k-1}} \min\{\lambda_{\min}(A_1 - \lambda I), \dots, \lambda_{\min}(A_k - \lambda I)\}. \quad (2.45)$$

Set $\alpha_i = p_{2i-1} + p_{2i}$, $i = 1, \dots, k$. We obtain

$$\operatorname{argmax}_{p \in \Delta_{2k-1}} \lambda_{\min}(A_i - \lambda I) = p_{2i}^{\text{opt}} = p_{2i-1}^{\text{opt}} = \frac{\alpha_i}{2},$$

so that (2.45) takes the form

$$p^{\text{opt}} = \operatorname{argmax}_{p \in \Delta_{2k-1}} \min\left\{\frac{\alpha_1(1-\rho_1)}{2}, \dots, \frac{\alpha_k(1-\rho_k)}{2}\right\}. \quad (2.46)$$

It is easy to verify that p^{opt} should satisfy

$$\frac{\alpha_1(1-\rho_1)}{2} = \dots = \frac{\alpha_k(1-\rho_k)}{2}.$$

The last relation leads to (2.13) and we conclude that the optimal selection probabilities p_i^{opt} , $i = 1, \dots, k$ are computed as in (2.14). Finally, (2.15) follows from (2.46).

□

Proof of Proposition 2. Note that the target function f in (2.18) is the minimum eigenvalue of

$$\begin{pmatrix} \sqrt{Q}D_w\sqrt{Q} & 0 \\ 0 & 1 - w_1 - \dots - w_s \end{pmatrix},$$

so that we can rewrite f as

$$\begin{aligned}
f(w) &= \lambda_{\min} (D_w^{\text{ext}} Q^{\text{ext}}) = \lambda_{\min} \left(\sqrt{Q^{\text{ext}}} D_w^{\text{ext}} \sqrt{Q^{\text{ext}}} \right) \\
&= \min \{ \lambda_{\min} (\sqrt{Q} D_w \sqrt{Q}), 1 - w_1 - \dots - w_s \} \\
&= \min \{ \text{P-Gap}(w), 1 - w_1 - \dots - w_s \}.
\end{aligned} \tag{2.47}$$

Let $w^* = \operatorname{argmax}_{w \in \Delta_s} f(w)$, and denote $p_j^* = \frac{w_j^*}{w_1^* + \dots + w_s^*}$, $j = 1, \dots, s$. To prove the proposition it suffices to show that for the pseudo-optimal weights p^{opt} ,

$$\text{P-Gap}(p^{\text{opt}}) \leq \text{P-Gap}(p^*). \tag{2.48}$$

Let $k^* = \sum_{i=1}^s w_i^*$. It is easy to see from (2.47) that

$$1 - k^* = \text{P-Gap}(w^*) = \text{P-Gap}(k^* p^*) = f(k^* p^*).$$

Since for any $k > 0$ and any $p \in \Delta_s$, $\text{P-Gap}(kp) = k \text{P-Gap}(p)$, we can choose $k < 1$ such that

$$1 - k = \text{P-Gap}(kp^{\text{opt}}) = f(kp^{\text{opt}}).$$

Hence, by definition of w^* ,

$$\text{P-Gap}(kp^{\text{opt}}) = 1 - k \leq 1 - k^* = \text{P-Gap}(k^* p^*),$$

implying $k^* \leq k$. Therefore,

$$\begin{aligned}
\text{P-Gap}(p^{\text{opt}}) &= \frac{1}{k} \text{P-Gap}(kp^{\text{opt}}) \leq \frac{1}{k} \text{P-Gap}(k^* p^*) \\
&= \frac{k^*}{k} \text{P-Gap}(p^*) \leq \text{P-Gap}(p^*),
\end{aligned}$$

whence we conclude (2.48).

□

Proof of Proposition 3. Note that $\left\langle \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x, x \right\rangle$ is linear in w for all $x \in \mathbb{R}^{d+1}$. That is, there exist functions $a_0(x), \dots, a_s(x)$ such that

$$\left\langle \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x, x \right\rangle = a_0(x) + w_1 a_1(x) + \cdots + w_s a_s(x).$$

Thus, $\left\langle \sqrt{Q_n^{\text{ext}}} D_n^{\text{ext}}(w) \sqrt{Q_n^{\text{ext}}} x, x \right\rangle$ is concave for all x . Then f_n is concave as the minimum over concave functions.

□

Proof of Proposition 4. Geometric ergodicity follows if we find drift coefficients to establish (A2). Let $x_{1:n} := (x_1, \dots, x_n)$. We argue that

$$V(x_{1:n}) = \frac{x_1^2}{2} + x_2^2 + \cdots + x_{n-1}^2 + \frac{x_n^2}{2}$$

is an appropriate drift function for the vanilla RSGS in cases (a), (b) and (c). Note that V does not depend on the regimes $r(i)$. One can work out the full conditionals for X_i ,

$$\begin{aligned} X_i | X_{-i}, Y_{1:n}, r(1), \dots, r(n) &\sim N(\mu, q_{r(i), r(i+1)}), \\ \mu &= q_{r(i), r(i+1)} \left(\frac{X_{i-1} I_{\{i>1\}}}{\sigma_{r(i)}^2} + \frac{X_{i+1} I_{\{i<n\}}}{\sigma_{r(i+1)}^2} + \frac{Y_i}{\beta^2} \right), \end{aligned} \quad (2.49)$$

where

$$q_{r(i), r(i+1)} = \frac{1}{\frac{1}{\beta^2} + \frac{I_{\{i>1\}}}{\sigma_{r(i)}^2} + \frac{I_{\{i<n\}}}{\sigma_{r(i+1)}^2}}.$$

Here we set $X_{n+1} = X_0 := 0$.

Let $f_i(x_{1:n}) = x_i^2$. For Pr_i , $i = 1, \dots, n$, defined in (2.1), where Pr_i corresponds for updating X_i from its full conditional, one sees it is obvious that

$$Pr_j f_i(x_{1:n}) = f_i(x_{1:n}), \quad i \neq j. \quad (2.50)$$

From (2.49), using the Cauchy-Schwartz inequality, we get

$$\begin{aligned} Pr_i f_i(x_{1:n}) &= \mu^2 + q_{r(i), r(i+1)} \\ &\leq q_{r(i), r(i+1)}^2 \left(\frac{I_{\{i>1\}}}{\sigma_{r(i)}^4} + \frac{I_{\{i<n\}}}{\sigma_{r(i+1)}^4} \right) \\ &\quad \times \left(f_{i-1}(x_{1:n}) I_{\{i>1\}} + f_{i+1}(x_{1:n}) I_{\{i<n\}} \right) + L_i(x_{i-1}, x_{i+1}), \end{aligned} \quad (2.51)$$

where $L_i(x_{i-1}, x_{i+1})$ is a linear function. Note that for the considered cases **(a)**, **(b)** and **(c)**, for any configuration of $(r(1), \dots, r(n))$ and $i \in \{2, \dots, n-1\}$,

$$2q_{r(i), r(i+1)}^2 \left(\frac{1}{\sigma_{r(i)}^4} + \frac{1}{\sigma_{r(i+1)}^4} \right) < 0.99. \quad (2.52)$$

Moreover, for $i \in \{1, n\}$,

$$q_{r(i), r(i+1)}^2 \left(\frac{I_{\{i>1\}}}{\sigma_{r(i)}^4} + \frac{I_{\{i<n\}}}{\sigma_{r(i+1)}^4} \right) \leq 0.57. \quad (2.53)$$

It follows from (2.51), (2.52) and (2.53),

$$\begin{aligned} & \frac{1}{2}Pr_1f_1(x_{1:n}) + \sum_{i=2}^{n-1} Pr_if_i(x_{1:n}) + \frac{1}{2}Pr_nf_n(x_{1:n}) \\ & \leq 0.99V(x_{1:n}) + L(x_{1:n}), \end{aligned} \quad (2.54)$$

where $L(x_{1:n})$ is a linear function. Together with (2.50), the inequality (2.54) yields

$$\frac{1}{d} \sum_{i=1}^n Pr_i V(x_{1:n}) \leq \frac{\lambda}{2} V(x_{1:n}) + A, \quad (2.55)$$

for some $\lambda < 1$ and $A < \infty$.

For Pr_{i+n} that corresponds for updating $r(i)$ from its full conditional, since V does not depend on $r(i)$, we get

$$Pr_{i+n}V = V. \quad (2.56)$$

Let $P_{\frac{1}{d}}$ be the RSGS kernel that corresponds to the vanilla chain with uniform sampling weights $\frac{1}{d}$. Combining (2.55) and (2.56) together, we obtain,

$$P_{\frac{1}{d}}V \leq \frac{\lambda}{2}V + \frac{1}{2}V + A.$$

Since for all $C < \infty$, set $\{(x_{1:n}, r(1), \dots, r(n)) \mid V(x_{1:n}) < C\}$ is small, Lemma 15.2.8 of [Meyn and Tweedie \[2009\]](#) yields that V is a geometric drift function.

For the RSGS with non-uniform selection probabilities $p = (p_1, \dots, p_d)$, we can use theorem 5 to conclude the geometric ergodicity.

□

Chapter 3

AirMCMC

As we discussed in Section 1.3.2, in this chapter we are interested in deriving the asymptotic convergence properties of

$$\hat{\pi}_N(f) := \frac{1}{N} \sum_{i=0}^{N-1} f(X_i) \quad (3.1)$$

for a class of AirMCMC Algorithms 4. In Section 3.1 we demonstrate performance of the Air version of the RWM algorithm. The main theoretical results are presented in Section 3.2. We compare our results with the available AMCMC theory in Section 3.3. Air versions of the ARSGS Algorithm 14 and the KAMH algorithm of Sejdinovic et al. [2014] are discussed in Section 3.4. We present detailed proofs for the main results in Section 3.6 with accompanying lemmas in Section 3.8. All other statements are proven in Section 3.7, which we clearly indicate.

3.1 Motivating Examples

In this section we examine the ability of AirMCMC to self tune, and see how it compares to standard Adaptive MCMC in its two most successful design versions that adapt the scaling and the covariance matrix of the proposal. We also empirically investigate sensitivity of AirMCMC to its key design parameter, the sequence of blocks lengths n_k .

3.1.1 Adaptive Scaling of Random Walk Metropolis

In this example we shall study Air version of the Adaptive Random walk Metropolis (ARWM) for a one dimensional target distribution. We consider an adaptive algorithm with normal proposals that tunes the proposal variance in order to achieve

the optimal acceptance ratio 0.44 (see Gelman et al. [1996]).

In Algorithm 16 we present Air version of the algorithm, where the adaptations of the variances are separated by the sequence of $\{n_k\}$ iterations. By taking $n_k \equiv 1$ we recover the original ARWM.

Below we compare performance of the ARWM with the AirRWM on sampling from a t-distribution.

$$\pi(x) \sim \left(1 + \frac{x^2}{\nu}\right)^{-(\nu+1)/2},$$

where we set $\nu = 10$ and consider three different sequences $n_k = \lfloor k^\beta \rfloor$ for $\beta \in \{1, 2, 3\}$. We start algorithms from with the initial proposal variance $\bar{\gamma} = (0.1)^2$. We also run a non-adaptive RWM with this initial variance to demonstrate the speed up of the adaptive algorithms.

Algorithm 16: AirRWM

Set some initial values for $X_0 \in \mathbb{R}$, $k := 1$, $n := 0$. Choose a slowly decaying to zero sequence $\{c_k\}_{k \geq 1}$.

Beginning of the loop

1. For $i = 1, \dots, n_k$

1.1. sample $Y \sim N(X_{n+i-1}, \bar{\gamma})$, $a_{\bar{\gamma}} := \min \left\{ \frac{\pi(Y)}{\pi(X_{n+i-1})}, 1 \right\}$;

1.2. $X_{n+i} := \begin{cases} Y & \text{with probability } a_{\bar{\gamma}}, \\ X_{n+i-1} & \text{with probability } 1 - a_{\bar{\gamma}}; \end{cases}$

1.3. $a := a + a_{\bar{\gamma}}$.

2. $\bar{\gamma} := \exp \left(\log(\bar{\gamma}) + c_k \left(\frac{a}{n_k} - 0.44 \right) \right)$.

3. Set $n := n + n_k$, $k := k + 1$, $a := 0$.

Go to **Beginning of the loop**

Remark. To prevent $\bar{\gamma}$ from converging to a poor proposal variance, the sequence c_k should be chosen so that $\sum_{i=1}^{\infty} c_k = \infty$, where N_i are the adaptation times. For example, we could choose $c_k := k^{-s}$ for some $s \in (0, 1)$.

Below we present the simulation results. The sequence c_k in the settings of the Algorithm 16 is chosen as in the above remark, $c_k := k^{-0.7}$. For every algorithm

we run 1000 independent chains for 100,000 iterations all started from the origin.

We estimate the optimal variance to be around 6.5. We observe that AirRWM with $\beta = 1$ approximates the optimal variance very well and performs only 446 adaptations; AirRWM with $\beta = 2$, performs 66 adaptations and underestimates the optimal variance to be 4.5; whereas in case $\beta = 3$, the AirRWM does only 24 iterations and estimates the variance only as 1.95. On the other hand, it is known that the adaptive algorithms are robust to the choice of the adapted parameters (see., e.g., [Gelman et al. \[1996\]](#)). As we can see in [Figure 3.1](#), all the adaptive algorithms estimate the 0.95 quantile equally well after 100,000 iterations. Note that the non-adaptive chain with proposal variance $(0.1)^2$ converges extremely slowly, so that its running quantile estimation plot does not fit [Figure 3.1](#). We present trace plots of the non-adaptive and adaptive chains in [Figure 3.2](#).

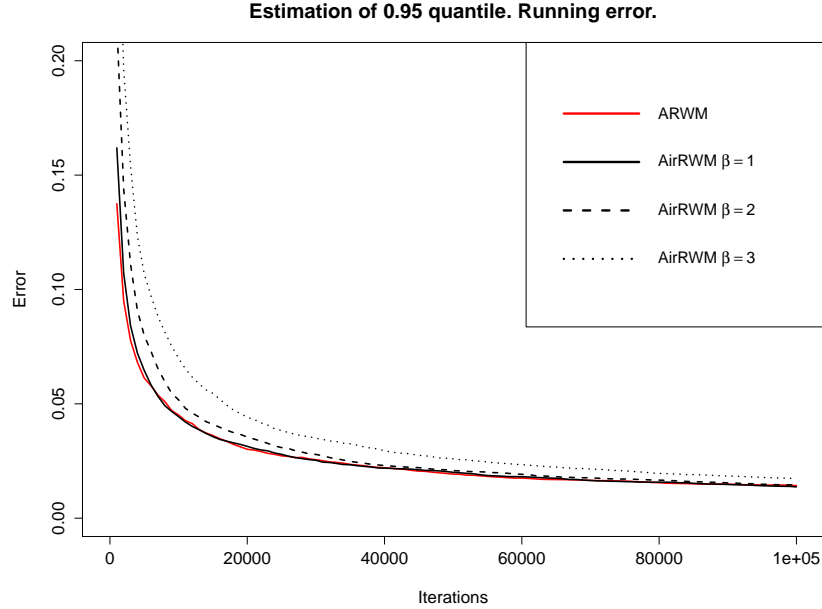


Figure 3.1: Error in estimation of a quantile at 0.95 level. X-axis – number of iterations. Y-axis – error in estimation

Remark. If the target distribution π has polynomial tails, then under mild conditions, as follows from results of Jarner and Roberts [Jarner and Roberts \[2007\]](#), the Random Walk Metropolis (RWM) with normal proposals produces a polynomially ergodic chain. More precisely, for some $r > 0$ consider a target distribution π on the whole line \mathbb{R} with Lebesgue density given by

$$\pi(x) = \frac{l(|x|)}{|x|^{1+r}}, \quad x \in \mathbb{R}, \quad (3.2)$$

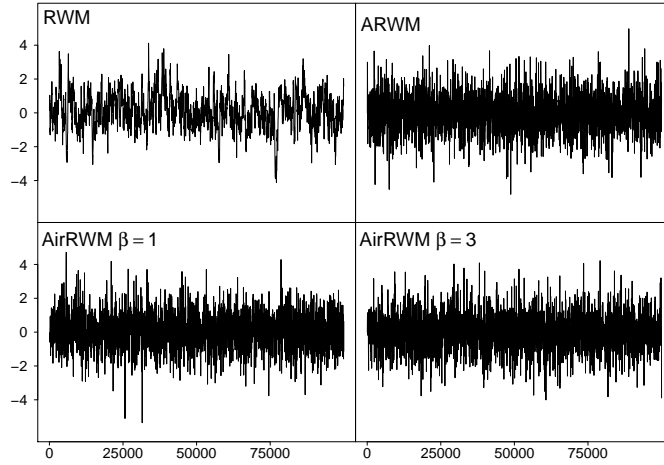


Figure 3.2: Trace plots

where $l(\cdot)$ is a normalised slowly varying function. By slowly varying function l we understand a function such that for all $a > 0$ $x^a l(x)$ is eventually increasing and $x^{-a} l(x)$ is eventually decreasing.

From Proposition 3 of [Jarner and Roberts \[2007\]](#), it follows that the collection of RWM kernels P_γ (here γ is a variance of the proposal) are simultaneously polynomially ergodic (see Assumption 3 in Section 3.2).

Thus, we can see that Theorem 12 of Section 3.2 is applicable and, given a sequence $\{n_k\}$ is chosen as in the theorem, the AirRWM Algorithm 16 produces a chain for which the SLLN and WLLN hold. If, additionally, the adapted variance $\bar{\gamma}$ converges, then the CLT holds, although we do not investigate further these details in the present chapter.

3.1.2 Adaptive Metropolis for high dimensional correlated posteriors

In this example we shall analyse ‘Air’ version of the Adaptive Random Walk Metropolis (ARWM) algorithm introduced by Haario et al. [Haario et al. \[2001\]](#) and studied in [Roberts and Rosenthal \[2009\]](#).

For a d -dimensional distribution π with covariance matrix Σ , consider a Metropolis-Hastings algorithm with a sequence of proposals

$$Q_n(x, \cdot) = 0.9N\left(x, \left[\frac{(2.38)^2}{d}\right]\Sigma_n\right) + 0.1N\left(x, \frac{(0.1)^2}{d}I_d\right),$$

where I_d is a d -dimensional identity matrix and Σ_n is a covariance matrix estimated from the first n steps of the adaptive algorithm.

The algorithm is aimed to approximate the optimal proposal $N\left(x, \frac{(2.38)^2}{d}\Sigma\right)$ (see [Roberts et al. \[1997\]](#), [Roberts and Rosenthal \[2001\]](#), [Rosenthal \[2011\]](#)), where Σ is a covariance matrix of the target distribution. Roberts & Rosenthal [Roberts and Rosenthal \[2009\]](#) argue that the ARWM may be very efficient in high-dimensional settings, where a good proposal is crucial. We shall analyse the same example as in Section 2 of [Roberts and Rosenthal \[2009\]](#). The target distribution is a multivariate normal

$$\pi \sim N(0, MM^T),$$

where the covariance matrix is formed of a $d \times d$ dimensional matrix with randomly generated entries $M_{ij} \sim N(0, 1)$.

For the ‘Air’ version of the algorithm, introduce a sequence of increasing lags,

$$n_k = \lfloor k^\beta \rfloor \quad k \geq 1,$$

for some $\beta > 0$, and consider Algorithm 4, where adaptations are allowed to take place only at times (1.6), i.e., after n_k non-adaptive iterations.

[Roberts and Rosenthal \[2009\]](#) measure the efficiency of an adaptive algorithm by looking at two crucial properties. First, is the ability of the algorithm to learn the appropriate scale (variance), which is monitored by looking at the trace plot. Second, is the ability of the algorithm to learn the shape of the target distribution, which is measured by *inhomogeneity factor* introduced by [Roberts and Rosenthal \[2001\]](#) (see also [Roberts and Rosenthal \[2009\]](#), [Rosenthal \[2011\]](#)). For a d -dimensional target distribution, the inhomogeneity factor is defined as

$$b = d \frac{\sum_{i=1}^d \lambda_i^{-1}}{\left(\sum_{i=1}^d \lambda_i^{-1/2}\right)^2},$$

where $\{\lambda_i\}_{i=1}^d$ are the eigenvalues of $\Sigma^{-1}\Sigma_n$, where, as before, Σ is the covariance matrix of π and Σ_n is the empirical covariance matrix. Note that by Jensen’s inequality, $b \geq 1$, and $b = 1$ only for the proposal, which shape is proportional to Σ .

For three different values of the parameter $\beta \in \{1, 3, 5\}$ we run ARWM and AirRWM algorithms to obtain 1 million samples for a 100 dimensional target distribution π . Trace plots of the 1st coordinate can be found in Figure 3.3, whereas the running inhomogeneity factor estimator is plotted in Figure 3.4.

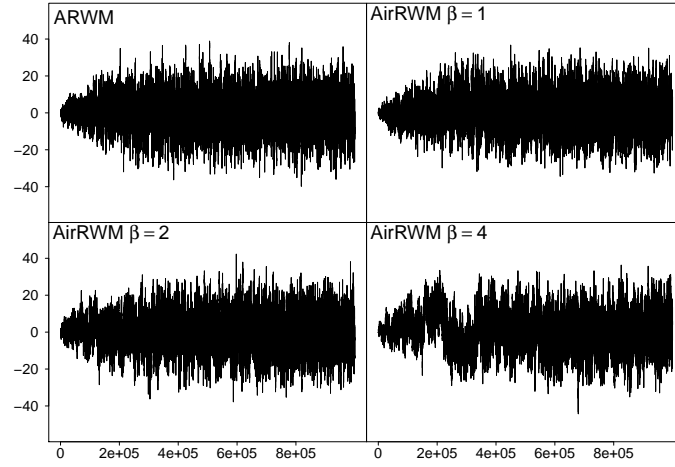


Figure 3.3: Trace of the 1st coordinate. $d = 100$

Surprisingly, it seems that AirRWM performs at least as well as the usual ARWM for any $\beta \in [1, 2]$, whence we conclude that one does not need to adapt the covariance matrix after each iteration. Moreover, we present total computational cost of the adaptive algorithm in Table 3.1. One can observe that Airing delivers a 5 fold speed up to the ARWM, where adaptations are performed at every iteration.

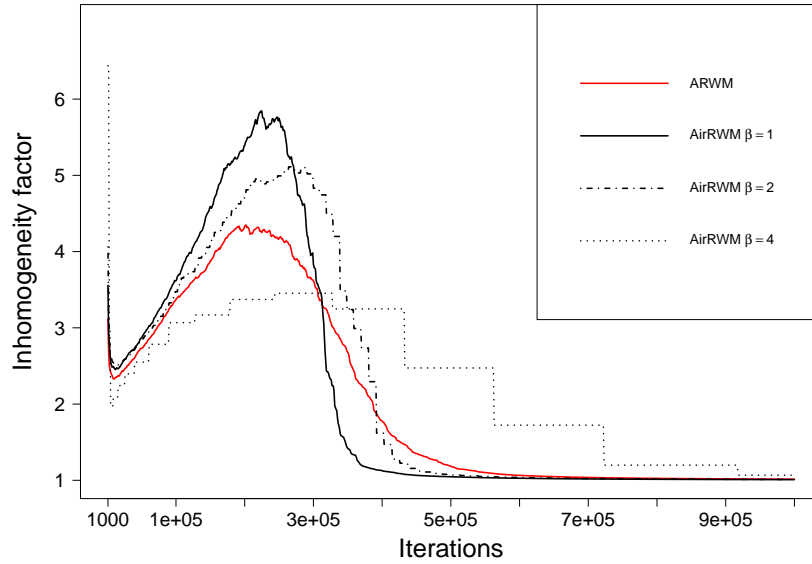


Figure 3.4: Inhomogeneity factor estimation. $d = 100$

Table 3.1: Time to obtain 1 million samples

	ARWM	AirRWM $\beta = 1$	AirRWM $\beta = 2$	AirRWM $\beta = 4$
Time (seconds)	507.6	90.5	86.9	80.2

3.2 AirMCMC Theory

Recall that we are interested in the long time behaviour of the sample average $\hat{\pi}_N(f)$ defined in (3.1), where the sequence $\{X_n\}_{n=0}^N$ is generated by the generic AirMCMC Algorithm 4. Hence, for n_{j+1} iterations between N_j and N_{j+1} , the process $\{X_n\}$ is evolving according to $P_{\gamma_{N_j}}$, and it is the properties of these Markov transition kernels that play the key role in the analysis.

The transition kernel P_γ , is a map $P_\gamma(\cdot, \cdot) : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \rightarrow [0, 1]$, such that $P_\gamma(x, \cdot)$ is a probability measure on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ for every $x \in \mathcal{X}$, and $P_\gamma(\cdot, A)$ is a $\mathcal{B}(\mathcal{X})$ measurable function for every $A \in \mathcal{B}(\mathcal{X})$. P_γ acts on the space of probability measures from the left, $\mu \rightarrow \mu P_\gamma$, with $\mu P_\gamma(A) := \int_{\mathcal{X}} P_\gamma(x, A) \mu(dx)$, and on the space of functions from the right, $f \rightarrow P_\gamma f$, with $P_\gamma f(x) := \int_{\mathcal{X}} f(y) P_\gamma(x, dy)$.

Given a collection of transition kernels $\{P_\gamma\}_{\gamma \in \Gamma}$, a sequence of lags $\{n_k\}_{k=0}^\infty$, an adaptation rule, say $R_{n+1} : \mathcal{X}^{n+2} \times \Gamma^{n+1} \rightarrow \Gamma$, and initialisation (X_0, γ_0) , the AirMCMC Algorithm 4 induces a probability measure on $\mathcal{X}^\infty \times \Gamma^\infty$, i.e. on the space of trajectories of $\{(X_n, \gamma_n)\}_{i=0}^\infty$. Denote this probability measure as $\mathbb{P}_{(X_0, \gamma_0)}$ and write $\mathbb{E}_{(X_0, \gamma_0)}$ for its expectation.

Properties of AirMCMC translate into statements about $\mathbb{P}_{(X_0, \gamma_0)}$ and, in particular,

- we say that the AirMCMC algorithm is *ergodic*, if it converges in distribution, i.e. for all $(X_0, \gamma_0) \in \mathcal{X} \times \Gamma$,

$$\lim_{n \rightarrow \infty} \|\mathcal{L}(X_n | X_0, \gamma_0) - \pi\|_{TV} = 0; \quad (3.3)$$

- the Mean Square Error of $\hat{\pi}_N(f)$ defined in (3.1) and obtained from AirMCMC, is

$$\text{MSE}(\hat{\pi}_N(f)) := \mathbb{E}_{(X_0, \gamma_0)} \left[\hat{\pi}_N(f) - \pi(f) \right]^2; \quad (3.4)$$

- the Weak Law of Large Numbers holds for AirMCMC, if for every $\varepsilon > 0$, $\hat{\pi}_N(f)$ converges in probability to $\pi(f)$, i.e.,

$$\lim_{N \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)}(|\hat{\pi}_N(f) - \pi(f)| > \varepsilon) = 0, \quad (3.5)$$

and we use \xrightarrow{P} to denote the convergence in probability;

- the Strong Law of Large Numbers holds for AirMCMC, if $\hat{\pi}_N(f)$ converges to $\pi(f)$ almost surely, i.e.,

$$\mathbb{P}_{(X_0, \gamma_0)} \left(\lim_{N \rightarrow \infty} \hat{\pi}_N(f) = \pi(f) \right) = 1, \quad (3.6)$$

and we use $\xrightarrow{a.s.}$ to denote almost sure convergence;

- and finally, the Central Limit Theorem holds if for every $u \in \mathbb{R}$,

$$\lim_{n \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)}(\sqrt{N}\{\hat{\pi}_N(f) - \pi(f)\} \leq u) = \frac{1}{\sqrt{2\pi\sigma_f^2}} \int_{-\infty}^u e^{-\frac{v^2}{2\sigma_f^2}} dv. \quad (3.7)$$

where $\sigma_f^2 = \sigma^2(f, P_\gamma) > 0$ is called the asymptotic variance. We use \xrightarrow{d} to denote convergence (3.7).

We start by introducing regularity conditions commonly used in analysis of MCMC and AMCMC algorithms. We refer to [Meyn and Tweedie \[2009\]](#), [Roberts and Rosenthal \[2004\]](#) for the Markov chains and MCMC context of these conditions, and to [Bai et al. \[2011\]](#), [Craiu et al. \[2015\]](#), [Roberts and Rosenthal \[2007\]](#) for the AMCMC context. Throughout the chapter the following will hold:

Assumption 1 (Regularity and Small Set).

- All considered Markov kernels P_γ are π -invariant, π -irreducible, and aperiodic (see [Meyn and Tweedie \[2009\]](#) for definitions);

- **One step simultaneous minorisation condition** holds, i.e., there exist a set $C \subseteq \mathcal{X}$, with positive mass $\pi(C) > 0$, a probability measure ν on $\mathcal{B}(\mathcal{X})$, and a constant $\delta > 0$, such that

$$P_\gamma(x, \cdot) \geq \delta \nu(\cdot) \quad \text{for all } x \in C, \gamma \in \Gamma. \quad (3.8)$$

We shall consider AirMCMC in several stability settings.

Assumption 2 (Simultaneous Geometric Drift). *The collection of transition kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ satisfies the Simultaneous Geometric Drift condition, if there exist constants $b < \infty$, $0 < \lambda < 1$, and a function $V : \mathcal{X} \rightarrow [1, \infty)$, such that*

$$P_\gamma V \leq \lambda V + bI_{\{C\}}, \quad \text{for all } \gamma \in \Gamma, \quad (3.9)$$

where C is the small set defined in (3.8).

Assumption 3 (Simultaneous Polynomial Drift). *The collection of transition kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ satisfies the Simultaneous Polynomial Drift condition, if there exist constants $b < \infty$, $0 < \alpha < 1$, $c > 0$, and a function $V : \mathcal{X} \rightarrow [1, \infty)$, such that*

$$P_\gamma V \leq V - cV^\alpha + bI_{\{C\}}, \quad \text{for all } \gamma \in \Gamma, \quad (3.10)$$

where C is the small set defined in (3.8).

Most theoretical work on Adaptive MCMC has been developed under simultaneous geometric or polynomial drift defined above, however these assumptions are not well suited for some classes of algorithms. Hence, we introduce the local simultaneous drift conditions.

Assumption 4 (Local Simultaneous Geometric Drift). *The collection of transition kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ satisfies the Local Simultaneous Geometric Drift condition, if for every $\gamma \in \Gamma$ there exist an open neighbourhood $B_\gamma \subseteq \Gamma$, such that $\gamma \in B_\gamma$, and there exist constants $b_\gamma < \infty$, $0 < \lambda_\gamma < 1$, and a function $V_\gamma : \mathcal{X} \rightarrow [1, \infty)$, such that*

$$P_{\gamma^*} V_\gamma \leq \lambda_\gamma V_\gamma + b_\gamma I_{\{C\}}, \quad \text{for all } \gamma^* \in B_\gamma, \quad (3.11)$$

where C is the small set defined in (3.8) for all P_γ , $\gamma \in \Gamma$.

The above formulation of the Local Simultaneous Drift condition is easy to verify in some fairly general settings, c.f. Theorem 5 of Chapter 2 for the case of Random Scan Gibbs Samplers. Recall, that Theorem 7 of Chapter 2 makes the Assumption 4 operational in a sense that it helps conclude ergodicity of the modified AMCMC Algorithms 13.

For the CLT to hold, we require a bound on the regeneration times of the Markov chain generated by a kernel P_γ . Assumption 1 allows construction of a split chain (X_n, Y_n) on the space $\mathcal{X} \times \{0, 1\}$ defined as

$$\begin{aligned}\mathbb{P}(Y_{n-1} = 1 | X_{n-1}) &= \delta I_{X_{n-1} \in C}, \\ \mathbb{P}(X_n \in A | Y_{n-1} = 1, X_{n-1}) &= \nu(A), \\ \mathbb{P}(X_n \in A | Y_{n-1} = 0, X_{n-1}) &= Q_\gamma(X_{n-1}, A),\end{aligned}$$

where

$$Q_\gamma(x, \cdot) = \frac{P_\gamma(x, \cdot) - \delta \nu(\cdot) I_{\{C\}}}{1 - \delta I_{\{C\}}}.$$

Note that marginally X_n is a Markov chain that evolves according to P_γ . Regeneration time $T = T(\gamma)$ is defined as

$$T = \inf\{n \geq 1 : Y_{n-1} = 1\}. \quad (3.12)$$

Assumption 5. For some $\delta > 0$ a function of interest f satisfies

$$\sup_{\gamma \in \Gamma} \mathbb{E}_{\nu, \gamma} \left[\sum_{j=0}^{T-1} f(X_j) \right]^{2+\delta} < \infty, \quad (3.13)$$

where $\mathbb{E}_{\nu, \gamma}$ is the expectation w.r.t. to a Markov chain with the kernel P_γ started from the measure ν (called regeneration measure), and $T = T(\gamma)$ is a regeneration time of a Markov chain with transition kernel P_γ .

For functions $g : \mathcal{X} \rightarrow \mathbb{R}$ and $V : \mathcal{X} \rightarrow [1, \infty)$ define a V -norm as

$$\|g\|_V := \sup_x \frac{|g(x)|}{V(x)}.$$

For a signed measure μ , the corresponding V -norm is defined as

$$\|\mu\|_V := \sup_{g: \|g\|_V=1} \|\mu(g)\|_V,$$

where $\mu(g) := \int g(x) d\mu(x)$.

Suppose that the parameter space Γ is a metric space. We say that the kernel P_γ is a continuous function of $\gamma \in \Gamma$ in V -norm if for any sequence $\{\gamma_n\}$ such that $\gamma_n \rightarrow \gamma$,

$$\sup_x \frac{\|P_{\gamma_n}(x, \cdot) - P_\gamma(x, \cdot)\|_V}{V(x)} \rightarrow 0 \text{ as } n \rightarrow \infty.$$

We are now ready to state the main results of the chapter.

3.2.1 Simultaneous Geometric Ergodicity

Theorem 10. *Let a collection of Markov kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ with an invariant distribution π satisfy Assumptions 1 and 2, and let (λ, V, b, C) be the drift coefficients in (3.9).*

Fix an arbitrary real number $\beta > 0$ and let $\{n_k\}_{k \geq 1}$ be a sequence such that for some $c_1 > 0$, $c_2 > 0$,

$$c_2 k^\beta \geq n_k \geq c_1 k^\beta. \quad (3.14)$$

For these parameters consider a chain $\{X_i\}_{i \geq 1}$ generated by the AirMCMC Algorithm 4.

Then for any starting distribution X_0 such that $\mathbb{E}V(X_0) < \infty$, and any function f such that $\|f\|_{V^{1/2}} := \sup_x \frac{|f(x)|}{V^{1/2}(x)} < \infty$:

- i) For any $\beta > 0$, the MSE of $\hat{\pi}_N(f)$ converges to $\pi(f)$ at a rate $N^{-\min\{1, \frac{2\beta}{1+\beta}\}}$, i.e.,*

$$\lim_{N \rightarrow \infty} \text{MSE}(\hat{\pi}_N(f)) = \mathcal{O}\left(\frac{1}{N^{\min\{1, \frac{2\beta}{1+\beta}\}}}\right),$$

in particular, the WLLN holds.

- ii) If $\beta \geq 1$, the rate in mean-square convergence is $\frac{1}{N}$, i.e.,*

$$\text{MSE}(\hat{\pi}_N(f)) = \mathcal{O}\left(\frac{1}{N}\right).$$

- iii) If $\beta > 1/2$, the SLLN holds,*

$$\hat{\pi}_N(f) \xrightarrow{a.s.} \pi(f).$$

iv) Suppose $\beta > 1$, Assumption 5 holds, Γ is a metric space, and the adapted parameter γ_{N_i} converges to a limit $\gamma_\infty \in \Gamma$ almost surely (where γ_∞ itself might be a random variable). Assume that P_γ is a continuous function of $\gamma \in \Gamma$ in $V^{1/2}$ -norm. If also, f has a positive asymptotic variance $\mathbb{P}_{(X_0, \gamma_0)}(\sigma^2(f, P_{\gamma_\infty}) > 0) = 1$, then the CLT holds, i.e.,

$$\sqrt{N}(\hat{\pi}_N(f) - \pi(f)) \xrightarrow{d} N(0, \sigma^2(f, P_{\gamma_\infty})).$$

Assumption 5 is standard to verify under simultaneous geometric ergodicity Assumption 2. We present the corresponding proposition below.

Proposition 5. *Let the Assumption 2 hold and $\sup_{x \in C} V(x) < \infty$. Then Assumption 5 holds for any function f such that $\|f\|_{V^{1/2-\delta}} < \infty$ for some $\delta > 0$.*

3.2.2 Local Simultaneous Geometric Ergodicity

In order to extend Theorem 10 to the local geometric ergodicity settings, we need to modify AirMCMC algorithm. We introduce a set B , where all the drift functions V_γ that satisfy (3.11), are bounded on B . Algorithm 17 is a modified version of AirMCMC, where the adaptations are allowed to take place only when the chain hits B .

Algorithm 17: Modified AirMCMC Sampler

Set some initial values for $X_0 \in \mathcal{X}$; $\gamma_0 \in \Gamma$; $\bar{\gamma} := \gamma_0$; $k := 1$; $n := 0$. Fix any set $B \in \mathcal{B}(\mathcal{X})$.

Beginning of the loop

1. For $i = 1, \dots, n_k$
 - 1.1. sample $X_{n+i} \sim P_{\bar{\gamma}}(X_{n+i-1}, \cdot)$;
 - 1.2. given $\{X_0, \dots, X_{n+i}, \gamma_0, \dots, \gamma_{n+i-1}\}$ update γ_{n+i} according to some adaptation rule.
2. Set $n := n + n_k$, $k := k + 1$. If $X_n \in B$, $\bar{\gamma} := \gamma_n$.

Go to Beginning of the loop

Remark. Efficiency of the algorithm depends on the choice of the set B . If B is too “small”, adaptations will not occur frequently. However under the conditions of

Theorem 11, the set B will be visited infinitely many times so that the adaptation will continue. Moreover, Theorem 7 of Chapter 2 implies that, if the parameter set Γ is compact, there exists a finite number of drift functions V_1, \dots, V_k that satisfy Assumption 4. Theorem 14.2.5. of Meyn and Tweedie [2009] implies that for large N , level sets $B = B(N) = \cap_{i=1}^k \{x : V_i(x) < N\}$, cover most of the support of π for large N , meaning that, with the appropriate choice of B , the adaptations will occur in most of the iterations of the modified Algorithm 17.

Theorem 11. *Let a collection of Markov kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ with an invariant distribution π satisfy Assumptions 1 and 4, and let $(\lambda_\gamma, V_\gamma, b_\gamma, C)$ be the drift coefficients in (3.9). Assume that Γ is a compact set in some topology and let $B \subset \mathcal{B}(\mathcal{X})$ be any set such that $\sup_{x \in B} V_\gamma(x) < \infty$, $\gamma \in \Gamma$.*

Fix an arbitrary real number $\beta > 0$ and let $\{n_k\}_{k \geq 1}$ be a sequence such that for some $c_1 > 0$, $c_2 > 0$,

$$c_2 k^\beta \geq n_k \geq c_1 k^\beta.$$

For these parameters consider the chain $\{X_i\}_{i \geq 1}$ generated by the AirMCMC algorithm 17.

Then for any starting distribution X_0 such that $\mathbb{E}V_\gamma(X_0) < \infty$, $\gamma \in \Gamma$, and any function f such that $\sup_x \frac{|f(x)|}{V_\gamma^{1/2}(x)} < \infty$, $\gamma \in \Gamma$:

- i) *For any $\beta > 0$, the MSE of $\hat{\pi}_N(f)$ converges to $\pi(f)$ at a rate $N^{-\min\{1, \frac{2\beta}{1+\beta}\}}$, i.e.,*

$$\lim_{N \rightarrow \infty} \text{MSE}(\hat{\pi}_N(f)) = \mathcal{O}\left(\frac{1}{N^{\min\{1, \frac{2\beta}{1+\beta}\}}}\right),$$

in particular, the WLLN holds.

- ii) *If $\beta \geq 1$, the rate in mean-square convergence is $\frac{1}{N}$, i.e.,*

$$\text{MSE}(\hat{\pi}_N(f)) = \mathcal{O}\left(\frac{1}{N}\right).$$

- iii) *If $\beta > 1/2$, the SLLN holds,*

$$\hat{\pi}_N(f) \xrightarrow{a.s.} \pi(f).$$

iv) Suppose $\beta > 1$, Assumption 5 holds, Γ is a metric space, and the adaptive parameter γ_{N_i} converges to a limit $\gamma_\infty \in \Gamma$ almost surely (where γ_∞ itself might be a random variable). Assume that for every $\gamma^* \in \Gamma$, P_γ is a continuous function of γ in some open neighbourhood of γ^* in $V_{\gamma^*}^{1/2}$ -norm. If also, f has a positive asymptotic variance

$\mathbb{P}_{(X_0, \gamma_0)}(\sigma^2(f, P_{\gamma_\infty}) > 0) = 1$, then the CLT holds, i.e.,

$$\sqrt{N}(\hat{\pi}_N(f) - \pi(f)) \xrightarrow{d} N(0, \sigma^2(f, P_{\gamma_\infty})).$$

The following proposition allows to practically verify Assumption 5 in the local geometric ergodicity settings.

Proposition 6. *Let the Assumption 4 hold and $\sup_{x \in C} V_\gamma < \infty$ for $\gamma \in \Gamma$. Then Assumption 5 holds for any function f such that $\|f\|_{V_\gamma^{1/2-\delta}} < \infty$ for some $\delta > 0$ and all $\gamma \in \Gamma$.*

3.2.3 Simultaneous Polynomial Ergodicity

In this section we extend Theorem 10 for the case of polynomially ergodic kernels P_γ , $\gamma \in \Gamma$.

Theorem 12. *Let a collection of Markov kernels $\{P_\gamma\}_{\gamma \in \Gamma}$ with an invariant distribution π satisfy Assumptions 1 and 3, and let (α, V, b, C, c) be the drift coefficients in (3.10). Assume also that $\alpha > 2/3$.*

Fix an arbitrary real number $\beta > 0$ and let $\{n_k\}_{k \geq 1}$ be a sequence such that for some $c_1 > 0$, $c_2 > 0$,

$$c_2 k^\beta \geq n_k \geq c_1 k^\beta.$$

For these parameters consider the chain $\{X_i\}_{i \geq 1}$ generated by the AirMCMC algorithm 4.

Then for any starting distribution X_0 such that $\mathbb{E}V(X_0) < \infty$, and any function f such that $\sup_x \frac{|f(x)|}{V^{\alpha/2-1}(x)} < \infty$:

i) *For any $\beta > \frac{\alpha}{4\alpha-2}$ the WLLN holds, i.e, for any $\varepsilon > 0$*

$$\lim_{N \rightarrow \infty} \mathbb{P}_{X_0, \gamma_0} \left(\left| \hat{\pi}_N(f) - \pi(f) \right| \geq \varepsilon \right) = 0.$$

ii) *If $\beta > 1/2 + \frac{\alpha}{4\alpha-2}$, the SLLN holds,*

$$\hat{\pi}_N(f) \xrightarrow{a.s.} \pi(f).$$

iii) Suppose $\beta > 1 + \frac{\alpha}{2\alpha-1}$, Assumption 5 holds, Γ is a metric space, and the adaptive parameter γ_{N_i} converges to a limit $\gamma_\infty \in \Gamma$ almost surely (where γ_∞ itself might be a random variable). Assume that P_γ is a continuous function of $\gamma \in \Gamma$ in $V^{\alpha 3/2-1}$ -norm. If also, f has a positive asymptotic variance $\mathbb{P}_{(X_0, \gamma_0)}(\sigma^2(f, P_{\gamma_\infty}) > 0) = 1$, then the CLT holds, i.e.,

$$\sqrt{N}(\hat{\pi}_N(f) - \pi(f)) \xrightarrow{d} N(0, \sigma^2(f, P_{\gamma_\infty})).$$

Remark. It follows from the theorem that $\beta > \frac{1}{2}$ disregarding the value of α .

As before, we present a proposition allows to practically verify Assumption 5 in simultaneous polynomial ergodicity settings.

Proposition 7. Let the Assumption 3 hold and $\sup_{x \in C} V < \infty$. Then Assumption 5 holds for any function f such that $\|f\|_{V^{\frac{\alpha(3\alpha-2)}{4\alpha-2}-\delta}} < \infty$ for some $\delta > 0$.

3.2.4 Convergence in distribution

We have shown in the previous section that under regularity conditions of Theorems 10, 11, and 12, the AirMCMC algorithm produces a chain with various convergence properties. However, without any additional assumptions the chain might fail to converge in distribution, as we demonstrate in Example 1 below. On the other, we show in Theorem 13 that imposing an additional *diminishing adaptation condition* (C1), guarantees ergodicity (i.e., convergence in distribution) of the AirMCMC algorithm. We argue that this is a minor condition that either holds in practice or can be easily enforced. In Theorem 14 we introduce an AirMCMC Algorithm 18, where the sequence of increasing lags $\{n_k\}$ is randomised, which ensures the diminishing adaptation condition.

Recall that the diminishing adaptation condition (C1) is a restriction on the adaptation size of the algorithm:

$$\sup_{x \in \mathcal{X}} \|P_{\gamma_n}(x, \cdot) - P_{\gamma_{n+1}}(x, \cdot)\|_{TV} \xrightarrow{P} 0 \text{ as } n \rightarrow \infty. \quad (3.15)$$

The following theorem demonstrates that the regularity conditions of the previous section together with the diminishing adaptation condition imply convergence in distribution of the AirMCMC algorithms.

Theorem 13. Suppose that the diminishing adaptation condition (3.15) and Assumption 1 hold. Let also one of the following conditions hold

- (a) Assumption 2 and for the corresponding drift function V , $\sup_{x \in C} V(x) < \infty$ and $\mathbb{E}V(X_0) < \infty$;
- (b) Assumption 4 and for the corresponding collection of drift functions V_γ , $\sup_{x \in C} V_\gamma(x) < \infty$ and $\mathbb{E}V_\gamma(X_0) < \infty$ for all $\gamma \in \Gamma$;
- (c) Assumption 3 and for the corresponding drift function V , $\mathbb{E}V(X_0) < \infty$ and every level set $C_d := \{x | V(x) \leq d\}$ is a uniform small set, i.e., satisfies (3.8).

Then any AirMCMC Algorithm 4 (or, in case (b), any modified AirMCMC Algorithm 17, where the set B in the algorithm settings is such that for the corresponding drift functions V_γ , $\sup_{x \in B} V_\gamma(x) < \infty$, $\gamma \in \Gamma$) produces an ergodic chain $\{X_n\}$, i.e.,

$$\|\mathcal{L}(X_n) - \pi\|_{TV} \rightarrow 0,$$

where $\mathcal{L}(X_n)$ is the distribution law of X_n .

As argued in Roberts and Rosenthal [2007], the diminishing adaptation condition is not an issue in practice. The condition holds for many typical adaptive MCMC algorithms (e.g., as for the standard Adaptive Metropolis or Adaptive Gibbs Samplers). For the adaptive algorithms where the condition does not hold (e.g., as for KAMH Sejdinovic et al. [2014]) or it is hard to verify the condition, we could, nevertheless, easily modify the algorithms in order to enforce (3.15). For example, at the adaptation times N_i , we could flip a coin with success probability p_i to decide whether to adapt the Markov kernel. If $\lim_{i \rightarrow \infty} p_i = 0$, then (3.15) holds. Notice that the sequence p_i can decay arbitrarily slowly.

Alternatively, for the AirMCMC algorithms, we could allow the sequence of increasing lags $\{n_k\}$ to be random. More precisely, let sequence $\{n_k^*\}$ be deterministic that satisfies (3.14) for some $\beta > 0$. We could consider an AirMCMC Algorithm 4, where in Step 2 we set $n_k = n_k^* + \text{Uniform}[0, \lfloor k^\kappa \rfloor]$ for some $\kappa \in (0, \beta)$. Since, $\{n_k\}$ satisfies (3.14), we could still prove the statements of Theorems 10, 11, 12 for this randomised version of the AirMCMC. Moreover, the resulting Algorithm 18 would be ergodic and satisfy the statements of Theorems 10, 11 or 12 under the corresponding regularity conditions. We summarise our observations in Theorem 14 below.

Theorem 14. Consider settings of Theorem 10 (alternatively, of Theorem 11 or 12), where the condition (3.14) holds for a sequence $\{n_k^*\}$. Consider an AirMCMC

Algorithm 18: Randomised AirMCMC Sampler

Set some initial values for $X_0 \in \mathcal{X}$; $\gamma_0 \in \Gamma$; $\bar{\gamma} := \gamma_0$. Let $\{n_k^*\}$ be an increasing sequence of positive integers. Fix some $\delta \in (0, 1)$. Set $k := 1$; $n := 0$, $n_1 := n_1^*$.

Beginning of the loop

1. For $i = 1, \dots, n_k$
 - 1.1. sample $X_{n+i} \sim P_{\bar{\gamma}}(X_{n+i-1}, \cdot)$;
 - 1.2. given $\{X_0, \dots, X_{n+i}, \gamma_0, \dots, \gamma_{n+i-1}\}$ update γ_{n+i} according to some adaptation rule.
2. Set $n := n + n_k$, $k := k + 1$, $n_k = n_k^* + \text{Uniform}[0, \lfloor k \rfloor^\delta]$, $\bar{\gamma} := \gamma_n$.

Go to **Beginning of the loop**

Algorithm 18 (in case of the settings of Theorem 11, we allow adaptations in Step 2 to happen only if the chain hits the corresponding set B). Then the adaptive chain $\{X_n\}$ produced by the algorithm satisfies statements of Theorem 10 (alternatively, of Theorem 11 or 12, respectively).

Moreover, for any sequence of lags $\{n_k^*\}$, the AirMCMC Algorithm 18 satisfies the diminishing adaptation condition (3.15). Under regularity conditions of Theorem 13 (in case of the settings (b) of the theorem, we allow adaptations to happen only if the chain hits the corresponding set B), the adaptive chain $\{X_n\}$ produced by the algorithm converges in distribution.

We conclude this section with a counterexample that demonstrates that an AirMCMC Algorithm 4 might fail to be ergodic (i.e., the corresponding adaptive chain does not converge in distribution) without the diminishing adaptation condition.

Example 1 This example is a modified version of Example 4 of Roberts & Rosenthal [2007]. Our goal is to construct an AirMCMC algorithm that satisfies conditions of Theorem 10 but fails to be ergodic. Let $\mathcal{X} = \{1, 2, 3, 4\}$. For some $\varepsilon > 0$, define a target as $\pi(\{1\}) := \varepsilon$, $\pi(\{2\}) := \varepsilon^3$, $\pi(\{3\}) = \pi(\{4\}) := \frac{1-\varepsilon-\varepsilon^3}{2}$. For $\gamma \in \Gamma := \{1, 2\}$, let P_γ correspond to a Metropolis-Hastings kernel with proposals

$$Q_1(x, \cdot) \sim \text{Uniform}\{x-1, x+1\}, \quad Q_2(x, \cdot) \sim \text{Uniform}\{x-2, x-1, x+1, x+2\}.$$

P_γ proceeds as follows. At every iteration given X_n , simulate proposal $Y_{n+1} \sim$

$Q_\gamma(X_n, \cdot)$, with probability $\min \left\{ 1, \frac{\pi(Y_{n+1})}{\pi(X_n)} \right\}$ set $X_{n+1} := Y_{n+1}$, otherwise, reject the proposal, i.e., $X_{n+1} := X_n$. If the proposal is outside \mathcal{X} , then we always reject it. Consider the following adaptive Algorithm 19.

Algorithm 19: AMCMC with the SLLN but failing convergence in distribution

Start with $X_0 = X_1 = X_2 = 1, \gamma_0 = \gamma_1 = \gamma_2 = 1, k = 1$.

Beginning of the loop

1. Sample $X_{2^{k^2}+1} \sim P_{\gamma_{2^{k^2}}}(X_{2^{k^2}}, \cdot)$;
2. If $\gamma_{2^{k^2}} = 1$, then update as follows. If $X_{2^{k^2}+1} \neq X_{2^{k^2}}$, i.e., the proposal is accepted, $\gamma_{2^{(k+1)^2}} := 2$. Otherwise, $\gamma_{2^{(k+1)^2}} := 1$;
If $\gamma_{2^{k^2}} = 2$, then set $\gamma_{2^{(k+1)^2}} = 1$ if $X_{2^{k^2}+1} = 1$, otherwise, set $\gamma_{2^{(k+1)^2}} = 2$.
3. For $n \in \mathbb{N}, 2^{k^2} + 2 \leq n \leq 2^{(k+1)^2}$ run a Markov chain with the kernel $P_{\gamma_{2^{(k+1)^2}}}$;
4. $k := k + 1$.

Go to **Beginning of the loop**

Proposition 8. *Kernels $P_\gamma, \gamma \in \Gamma$ satisfy the simultaneous minorisation Assumption 1 and the simultaneous drift Assumption 2. Therefore, by virtue of Theorem 10, the SLLN holds for Algorithm 19. However, the adaptive chain produced by the algorithm does not converge in distribution for sufficiently small $\varepsilon > 0$ in the setting of the underlying target distribution.*

Proof of Proposition 8. Algorithm 19 is designed in such a way, that Steps 1 and 2 “drift” the adaptive $\{X_n\}$ chain away from the correct stationary distribution. Since the chain approaches the stationary distribution arbitrarily closely after Step 3, we conclude that at times $2^{k^2} + 2$,

$$\left\| \mathcal{L} \left(X_{2^{k^2}+2} | X_0, \gamma_0 \right) - \pi \right\|_{TV} > \delta$$

for some $\delta > 0$. Therefore, $\{X_n\}$ does not converge in distribution. We provide a detailed proof of the proposition in Appendix A.

□

3.3 Comparison with available Adaptive MCMC theory

AMCMC algorithms have received an increasing attention in the past two decades with much research devoted to studying ergodicity property [Atchadé and Rosenthal \[2005\]](#), [Bai et al. \[2011\]](#), [Haario et al. \[2001\]](#), [Łatuszyński et al. \[2013b\]](#), [Roberts and Rosenthal \[2007\]](#), robustness and stability of the algorithms [Andrieu and Atchadé \[2007\]](#), [Craiu et al. \[2015\]](#), [Vihola \[2012\]](#), as well as asymptotic behaviour of the average (3.1) of the adaptive chain output [Andrieu and Moulines \[2006\]](#), [Atchadé and Fort \[2010\]](#), [Gilks et al. \[1998\]](#), [Saksman and Vihola \[2010\]](#), [Vihola \[2011\]](#). In the current paper we are interested in the latter part, i.e., in studying the asymptotic behaviour of (3.1).

As discussed in [Roberts and Rosenthal \[2007\]](#), convergence of any AMCMC algorithm depends on the combination of two factors: the speed of convergence of the underlying Markov kernels P_γ (their mixing properties) and the adaptation scheme of the algorithm. In practice the appropriate combination of mixing and adaptation is established by verifying the *containment* and *diminishing adaptation* conditions. Together these conditions imply convergence in distribution of the AMCMC (see [Roberts and Rosenthal \[2007\]](#)). Violating either of the conditions can ruin the convergence of an AMCMC scheme (see e.g., example in Section 3 of [Łatuszyński et al. \[2013b\]](#); Examples 1 and 2 in [Roberts and Rosenthal \[2007\]](#)). As we discussed in Section 3.2.4, the diminishing adaptation is a mild condition, that can be imposed, if necessary, by slightly modifying the adaptation procedure.

The containment condition is not necessary for convergence, but an ergodic adaptive algorithm that fails the containment, is also more inefficient than any of its non-adaptive counterparts, as was proven in [Łatuszyński and Rosenthal \[2014\]](#). The containment is a technical condition, which is notoriously hard to verify directly. However, it is implied by the regularity assumptions presented in Section 3.2 (see [Bai et al. \[2011\]](#), [Roberts and Rosenthal \[2007\]](#), and Theorem 7 of Chapter 2). As demonstrated in Example 4 [Roberts and Rosenthal \[2007\]](#), even on finite state spaces the containment and diminishing adaptation conditions alone do not guarantee the SLLN.

Under the diminishing adaptation and simultaneous geometric drift condition (3.9), the SLLN was established in, e.g., [Andrieu and Moulines \[2006\]](#), [Atchadé and Fort \[2010\]](#), [Saksman and Vihola \[2010\]](#), [Vihola \[2011\]](#). Moreover, under an additional assumption that the adapted parameters converge, the CLT was established in [Andrieu and Moulines \[2006\]](#). The SLLN was also established under the simultaneous polynomial drift condition (3.10) in [Atchadé and Fort \[2010\]](#). Note,

however, the authors effectively require the joint process (X_n, γ_n) to be an inhomogeneous Markov chain. The results are well-suited for many popular algorithms, e.g., Adaptive Metropolis-Hastings (see Section 3.2 in [Atchadé and Fort \[2010\]](#)), Adaptive Metropolis-within-Gibbs (see, e.g., [Łatuszyński et al. \[2013b\]](#), [Rosenthal \[2011\]](#)), or Adaptive Metropolis adjusted Langevin Algorithm (see [Atchadé \[2006\]](#)).

On the other hand, there are algorithms that do not meet the conditions of [Andrieu and Moulines \[2006\]](#), [Atchadé and Fort \[2010\]](#), [Saksman and Vihola \[2010\]](#), [Vihola \[2011\]](#). For example, the ARSGS Algorithm 9 presented in Chapter 2, generally does not satisfy the simultaneous drift condition, whereas the Kernel Adaptive Metropolis-Hastings (KAMH) algorithm, proposed by [Sejdinovic et al. \[2014\]](#), produces an adaptive chain (X_n, γ_n) that is not Markov. Furthermore, none of the available adaptive MCMC results quantifies the MSE rate of convergence.

We have introduced a concept of AirMCMC algorithms, for which we have relaxed the generally imposed conditions. First, we do not require the joint adaptive chain (X_n, γ_n) to be Markov. Secondly, for the modified AirMCMC Algorithm 17, instead of the simultaneous geometric drift condition (3.9), we require only the local geometric drift (3.11) to hold, which is a natural condition for the ARSGS. Thus, we could prove the SLLN, MSE convergence, and convergence in distribution for the Air versions of the ARSGS and the KAMH in Section 3.4.

Moreover, for the AirMCMC algorithms, under the local geometric drift Assumption 4 or the simultaneous polynomial drift Assumption 3, we have established the CLT. We have also derived the MSE convergence under the local or simultaneous geometric drift conditions (Assumptions 4 and 2, respectively).

We emphasise that virtually any AMCMC algorithm can be transformed into an Air version via lagging the adaptations in a way described in Algorithms 4, 17, and 18.

The technique we have used for analysis is tightly related to the one developed by [Gilks et al. \[1998\]](#). The key idea in [Gilks et al. \[1998\]](#) is to allow adaptations of the Markov kernel P_γ to happen only at suitably constructed regeneration times of the chain. Under only Assumption 1, it is then possible to establish the SLLN, CLT, and MSE convergence. This is an effective idea for AMCMC in low dimensional spaces but impractical in higher dimensions, since the chain typically regenerates at a rate which recedes to 0 exponentially in dimension.

By introducing an increasing sequence of iteration $\{n_k\}$ between adaptation in Algorithms 4, 17, and 18, we have shown that the regularity conditions of Section 3.2 guarantee that the chain regenerates between adaptations with an increasingly

high probability. Since $\{n_k\}$ grows sufficiently fast, we can use the technique of Gilks et al. [1998] to analyse the Markov tours of the adaptive chain between the regenerations, and control the remainder terms of the adaptive chain using the explicit bounds of Latuszyński et al. [2013a].

3.4 Examples: Air versions of complex AMCMC algorithms

3.4.1 Adaptive Random Scan Gibbs Sampler

We could directly apply Theorem 11 to the ARSGS Algorithm 14 presented in Chapter 2. Let $p = (p_1, \dots, p_s)$ be a probability vector and assume that the target distribution sits on a product space $\mathcal{X}_1 \times \dots \times \mathcal{X}_s$. Recall, that the RSGS proceeds at each iteration by first choosing a coordinate i with probability p_i , and then updating the coordinate from its full conditional distributions. In the ARSGS Algorithms 9 and 14 the adaptations of the selection probabilities p are separated by k_i RSGS iterations. Therefore, if the sequence k_i is chosen to be non-decreasing, the ARSGS already fits into the AIRMCMC framework.

As we mentioned in Section 3.3, it is hard to verify the simultaneous geometric drift condition (3.9) for the ARSGS. On the other hand, the local simultaneous geometric drift condition (3.11) is a natural property for the ARSGS as long as the RSGS Markov kernel is geometrically ergodic for at least some selection probability vector $p = (p_1, \dots, p_s)$ (see Theorem 5 of Chapter 2). We summarise our observations in the following theorem

Theorem 15. *Let π be a target distribution on $\mathcal{X}_1 \times \dots \times \mathcal{X}_s$, where $\mathcal{X}_i = \mathbb{R}^{d_i}$ for some positive integers d_1, \dots, d_s . Consider a collection of RSGS kernels P_p parametrised by the sampling weights $p = (p_1, \dots, p_s)$. Assume that P_p satisfy Assumption 1 and for some $p = (p_1, \dots, p_s)$, P_p is geometrically ergodic, i.e., (3.9) holds. Then:*

1. *The collection of kernels P_p satisfy the local simultaneous drift condition (3.11).*
2. *Then the modified ARSGS Algorithm 14 described in Chapter 2, with the corresponding sequence of lags between adaptations $k_i = \lfloor ci^\beta \rfloor$ for some $\beta > 0$, $c > 0$, is an example of AirMCMC algorithm for which i) - iii) of Theorem 11 hold.*

Proof of Theorem 15. Part 1 follows from Theorem 5 of Chapter 2. Part i) of the theorem follows by simple application of Theorem 11.

□

Remark. One needs the adapted selection probabilities to converge, in order to derive the CLT using iv) of Theorem 11. We do not have a proof that the adapted selection probabilities converge at all. However, one could choose the learning rate a_m in the settings of the ARSGS so that the adapted probabilities converge to a suboptimal value (i.e., take a_m such that $\sum_{m=1}^{\infty} a_m < \infty$, where a_m is defined in the settings of Algorithm 9 and used as a learning rate in Step 3.2 of the algorithm). Now we are in a position to use iv) of Theorem 11 in order to verify the CLT.

3.4.2 Kernel Adaptive Metropolis-Hastings

Our results are also applicable to the Kernel Adaptive Metropolis-Hastings (KAMH) algorithm presented in [Sejdinovic et al. \[2014\]](#). The idea behind the KAMH is to locally adapt the variance of a symmetric random walk proposal based on a subsample of the whole previous chain history. Thus, the adaptive chain (X_n, γ_n) is not Markovian so that the results of [Andrieu and Atchadé \[2007\]](#), [Atchadé and Fort \[2010\]](#) do not apply. However, one may easily put the algorithm into the Air framework. We shall provide conditions which ensure that i) - iii) of Theorem 10 hold for the AirKAMH and thus, establish the SLLN and MSE convergence for the algorithm.

KAMH is an Adaptive Metropolis algorithm with a family of local proposals

$$Q_{Z,\nu}(x, \cdot) = N(x, \kappa I + \nu^2 M(x, Z)), \quad (3.16)$$

where $M(x, Z)$ is a $d \times d$ positive semidefinite matrix that depends on a current position $x \in \mathbb{R}^d$ and $d \times t$ matrix Z (see (3.17) for the precise representation). Here each column Z_i , $i = 1, \dots, t$ of Z is a randomly chosen state from the adaptive chain history, γ is a fixed scale parameter (e.g., $\kappa = 0.2$), and ν is tuned on the fly in order to retain the average acceptance ratio around 0.234 (see e.g., [Andrieu and Thoms \[2008\]](#), [Rosenthal \[2011\]](#), [Roberts et al. \[1997\]](#)). Let $\{p_i\}$ be a sequence of probability weights slowly decaying to zero. Let $q_{Z,\nu}$ be the density corresponding to (3.16). The KAMH proceeds by iterating through three steps:

1. With probability p_n , subsample $Z = (Z_1, \dots, Z_t)$ from the whole current output $\{X_1, \dots, X_n\}$;
2. Generate a proposal Y from (3.16);

3. Accept/reject the proposal using the standard Metropolis acceptance ratio

$$\alpha(X_n, Y) = \min \left\{ 1, \frac{\pi(Y)q_{Z,\nu}(X_n, Y)}{\pi(X_n)q_{Z,\nu}(Y, X_n)} \right\}.$$

4. Tune the proposal variance ν to retain the average acceptance ratio around 0.234:

$$\nu := \exp \left(\log(\nu) + \frac{1}{\sqrt{n}} \{ \alpha(X_n, Y) - 0.234 \} \right).$$

Implicitly $M(x, Z)$ depends on a covariance kernel $k(x, y)$ in \mathbb{R}^d :

$$\begin{aligned} M(x, Z) &= V(x, Z) \left(I_t - \frac{1}{t} \mathbb{1}_t \right) V^\top(x, Z), \\ V(x, Z) &= 2(\nabla_x k(x, z_1), \dots, \nabla_x k(x, z_t)), \end{aligned} \tag{3.17}$$

where I_t is a $t \times t$ identity matrix and $\mathbb{1}_t$ is a $t \times t$ matrix of ones. If $k(x, y)$ is a linear kernel (i.e., $k(x, y) = x^\top y$), then $M(x, Z) = M(Z)$ does not depend on x and approximates the global covariance structure of the target distribution. More complicated kernels $k(x, y)$, e.g., the Gaussian or Matérn kernel, (see [Sejdinovic et al. \[2014\]](#) for the definitions), $Q_{z,\nu}(x, \cdot)$ allow for local approximation of the covariance structure. Thus, KAMH has the potential to adapt to distributions with complicated shapes.

Below we shall show, if the target distribution has super-exponential tails one can establish the simultaneous geometric ergodicity Assumption 2, if (Z, ν) are restricted to any compact domain.

Proposition 9. *Assume that the target distribution π in \mathbb{R}^d has a density w.r.t. Lebesgue measure, which is differentiable, bounded, has super-exponential tails, i.e.,*

$$\limsup_{|x| \rightarrow \infty} \left\langle \frac{x}{|x|}, \nabla \log \pi(x) \right\rangle = -\infty,$$

and satisfies the curvature condition

$$\limsup_{|x| \rightarrow \infty} \left\langle \frac{x}{|x|}, \frac{\nabla \log \pi(x)}{|\nabla \log \pi(x)|} \right\rangle < 0,$$

where $|\cdot|$ and $\langle \cdot, \cdot \rangle$ are the norm and the scalar product in \mathbb{R}^d respectively. Let $k(x, y)$ be a Gaussian or Matérn kernel. Then the collection of Metropolis kernels $\{P_{Z,\nu}\}_{(Z,\nu) \in \Gamma}$ with the corresponding proposals $\{Q_{Z,\nu}(x, \cdot)\}_{(Z,\nu) \in \Gamma}$, satisfy Assump-

tion 1 and the simultaneous geometric drift Assumption 2 for any compact set Γ in $\mathbb{R}^{d \times t+1}$.

Proof of Proposition 9. See Appendix A.

□

For the Air version of the KAMH, we update Z in Step 1 at the pre-specified times (1.6), N_i , whereas the proposal ν in Step 4 could be updated at the times $\lfloor \frac{N_i}{l} \rfloor$ for some integer $l \geq 1$, in the same manner as in Algorithm 16 of Section 3.1.

Theorem 16. *Assume that the target distribution is super-exponentially tailed, differentiable, bounded, and (Z, ν) are restricted to any compact domain $\Gamma \subset \mathbb{R}^{d \times t+1}$. Then for an Air version of the KAMH, i) - iii) of Theorem 10 hold.*

Proof of Theorem 16. Follows from Proposition 9.

□

Remark. One can see that due to Step 1 of the KAMH, the adapted parameter $\gamma = (\nu, Z)$ does not converge, since we randomly subsample Z infinitely often. Thus, we can not apply iv) of Theorem 10 to derive the CLT.

3.5 Discussion

In this chapter we introduced a class of AMCMC algorithms, AirMCMC, where adaptations are separated with a sequence of increasing lags $\{n_k\}$. In Section 3.2 we have proved that the simultaneous or local simultaneous drift Assumptions 2 or 4, imply the SLLN, MSE convergence and, if the adapted parameter converges, the CLT for the AirMCMC. The same technique was used to prove the SLLN and CLT under the simultaneous polynomial drift Assumption 3.

In Sections 3.1 and 3.4 we have demonstrated that many of the known AMCMC can be put into the Air framework (Algorithms 4 and 17). In Section 3.4 we have seen that this could lead to the algorithms with theoretical underpinning for the asymptotic convergence properties of the averages (3.1). Moreover, empirically, in Section 3.1 we have demonstrated that including a lag between the adaptations does not necessarily slow down convergence of the adaptive algorithm. On the contrary, in Section 3.1.2, we have experienced computational speed up, since the Air version of the adaptive algorithm spent less time adapting the parameter.

Our settings are different from what we have seen in the literature since the diminishing adaptation condition (3.15) does not necessarily hold. As we have

seen in Section 3.2.4, without the diminishing adaptation condition, the AirMCMC algorithm might converge in distribution. This does not affect the properties of ergodic averages (3.1), and also it is easy to impose the condition, which guarantees convergence in distribution, as we have proven in Theorem 13 of Section 3.5.

We have discussed in Section 3.3 that our settings are closely related to the ones of Gilks et al. [1998], where the authors consider AMCMC with adaptations allowed to happen only at the regeneration times of the underlying Markov chains. It follows, that in the settings of Gilks et al. [1998], one can establish the MSE convergence and the CLT of the AMCMC. Unfortunately, the framework of Gilks et al. [1998] is not useful in high dimensional settings, since the regeneration times deteriorate to zero exponentially in dimension. On the other hand, by introducing a sequence of increasing lags $\{n_k\}$ between adaptations, that grow sufficiently fast, the underlying Markov chains between the adaptations regenerate with an increasing to 1 probability, which allows us to exploit technique of Gilks et al. [1998] in the proofs of the main results.

An important open question about the design of AirMCMC algorithms is the optimal choice of the sequence $\{n_k\}$ that could potentially be established through information theoretical arguments (see MacKay [2003]).

3.6 Proofs for Section 3.2

In this section we prove the theorems and propositions from Section 3.2. We first prove Theorems 10, 11 and 12. The rest of the results are proven in the same order they appear in the paper. Accompanying lemmas are proven in Appendix B.

We start with the general approach valid for any of the Theorems 10, 11, 12. Without loss of generality we assume $\pi(f) = 0$. As before, $N_0 = 0$, $N_i = N_{i-1} + n_i$. The following lemma provides the rate of growth of N_k relative to k .

Lemma 6. *For all $\beta > 0$ and $n \geq 1$,*

$$\sum_{i=1}^n i^\beta = \frac{1}{1+\beta} n^{1+\beta} + o(n^{1+\beta}), \text{ as } n \rightarrow \infty.$$

It follows from Lemma 6, and the assumption (3.14), that for some $\hat{c} > 0$,

$$\frac{1}{\hat{c}} k^{1+\beta} \geq N_k \geq \hat{c} k^{1+\beta}. \quad (3.18)$$

For $i \geq 1$ define

$$s_i(f) = \sum_{j=N_{i-1}}^{N_i-1} f(X_j).$$

For each i consider a Markov chain $\{Y_j^{(i)}\}$ with a kernel $P_{\gamma_{N_{i-1}}}$ started at $X_{N_{i-1}}$, such that for $j \in \{0, \dots, n_i - 1\}$,

$$Y_j^{(i)} := X_{N_{i-1}+j}, \quad (3.19)$$

and for $j \geq n_i$, $\{Y_j^{(i)}\}$ evolves independently of $\{X_{N_i}, X_{N_i+1}, \dots\}$.

Let $T_k^{(i)}$ be the k -th regeneration time (see (3.12) for the definition) of the chain $Y_j^{(i)}$. Set

$$T^{(i)} := T_1^{(i)}$$

and

$$R_i(n) := \inf\{r \geq 1 : T_r^{(i)} \geq n\}.$$

For $i, j \geq 1$ define

$$\begin{aligned} \eta_i(f) &= \sum_{j=0}^{T^{(i)}-1} f(Y_j^i), & \xi_i(f) &= \sum_{j=T^{(i)}}^{T_{R_i(n_i)}^{(i)}-1} f(Y_j^i), \\ \zeta_i(f) &= \sum_{j=n_i}^{T_{R_i(n_i)}^{(i)}-1} f(Y_j^i), & \xi_{i,j}(f) &= \sum_{m=T_j^{(i)}}^{T_{j+1}^{(i)}-1} f(Y_m^i). \end{aligned}$$

where $\xi_i(f) := 0$ if $T^{(i)} = T_{R_i(n_i)}^{(i)}$.

The partial sum $s_i(f)$ can be represented as

$$s_i(f) = \eta_i(f) + \xi_i(f) - \zeta_i(f).$$

For the average

$$S_N(f) := \sum_{j=0}^N f(X_j)$$

find $k = k(N)$ such that $N_k < N < N_{k+1}$. We shall rewrite S_n as a sum of four term each of which we analyse separately.

$$\begin{aligned}
S_N(f) &= \sum_{i=1}^k s_i(f) + \sum_{j=N_k}^N f(X_i) \\
&= \sum_{i=1}^k \eta_i(f) + \sum_{i=1}^k \xi_i(f) - \sum_{i=1}^k \zeta_i(f) + \sum_{j=N_k}^N f(X_i) \\
&= \Xi_{N_k}^{(1)} + \Xi_{N_k}^{(2)} + \Xi_{N_k}^{(3)} + \Xi_{N_k, N}^{(4)}.
\end{aligned} \tag{3.20}$$

Terms $\Xi_{N_k}^{(i)}$, $i \in \{1, 3\}$ and $\Xi_{N_k, N}^{(4)}$ will be analysed later below with using specific conditions of every theorem.

On the contrary, the main term $\Xi_{N_k}^{(2)}$, containing most of the adaptive chain trajectory, can be analysed similarly for all the theorems using the standard renewal theory approach as suggested by [Gilks et al. \[1998\]](#). We prove properties of $\Xi_{N_k}^{(2)}$ in the following proposition.

Proposition 10. *Suppose that the conditions of either Theorem 10, 11 or 12 hold. Then*

$$\mathbb{E}_{(X_0, \gamma_0)} \left[\frac{1}{N_k} \Xi_{N_k}^{(2)} \right]^2 = \mathcal{O} \left(\frac{1}{N_k} \right). \tag{3.21}$$

Assume also that the CLT asymptotic variance $\sigma^2(f, P_\gamma)$ is a continuous function of γ and $\gamma_{N_k} \rightarrow \gamma_\infty \in \Gamma$. If also, $\sigma_\infty^2 := \sigma^2(f, P_{\gamma_\infty}) > 0$, then

$$\frac{1}{\sqrt{N_k}} \Xi_{N_k}^{(2)} \xrightarrow{d} N(0, \sigma_\infty^2). \tag{3.22}$$

Proof of Proposition 10. First, note that simultaneous minorisation condition (3.8) yields that

$$\mu := \mathbb{E}_{\nu, \gamma} T \tag{3.23}$$

is independent of γ , since $\mathbb{E}_{(\nu, \gamma)} T = \frac{1}{\delta\pi(C)}$ (see (3.3.6) and (3.5.2) of [Nummelin \[2002\]](#)).

Note that $\xi_i(f)$ can be written as

$$\xi_i(f) = \sum_{j=1}^{R_i(n_i)-1} \xi_{i,j}(f).$$

Introduce a filtration

$$\mathcal{F}_0 = \{\emptyset\}, \mathcal{F}_i = \sigma \left\{ \mathcal{F}_{i-1} \cup \left\{ Y_0^{(i)}, \dots, Y_{T_{R_i(n_i)}^{(i)}-1}^{(i)} \right\} \right\}. \quad (3.24)$$

The sequence $\{\xi_i\}$ is adapted to \mathcal{F}_i . Note that conditionally on \mathcal{F}_{i-1} , variables $\{(\xi_{i,j}, T_{j+1}^{(i)} - T_j^{(i)})\}_{j \geq 1}$ are i.i.d. as tours between regenerations of a Markov chain. Therefore, we can use first Wald's identity in order to get the following representation:

$$\mathbb{E}_{(X_0, \gamma_0)} [\xi_{i+1} | \mathcal{F}_i] = E_{(X_0, \gamma_0)} [\xi_{i+1,1} | \mathcal{F}_i] \mathbb{E}_{(X_0, \gamma_0)} [R_{i+1}(n_{i+1}) - 1 | \mathcal{F}_i]. \quad (3.25)$$

and use relations (3.3.7), (3.5.1) of Nummelin [2002] to see that

$$E_{(X_0, \gamma_0)} [\xi_{i+1,j} | \mathcal{F}_i] = \pi(f) \mu. \quad (3.26)$$

Therefore, since $\pi(f) = 0$ by the assumption, (3.25) and (3.26) imply

$$\mathbb{E}_{(X_0, \gamma_0)} [\xi_{i+1} | \mathcal{F}_i] = 0,$$

whence

$$\mathbb{E}_{(X_0, \gamma_0)} [\xi_i \xi_{i+1}] = \mathbb{E}_{(X_0, \gamma_0)} [E[\xi_i \xi_{i+1} | \mathcal{F}_i]] = \mathbb{E}_{(X_0, \gamma_0)} [\xi_i E[\xi_{i+1} | \mathcal{F}_i]] = 0.$$

We conclude that for $i \neq j$,

$$\mathbb{E}_{(X_0, \gamma_0)} [\xi_i \xi_j] = 0.$$

It follows,

$$\mathbb{E}_{(X_0, \gamma_0)} \left[\left(\Xi_{N_k}^{(2)} \right)^2 \right] = \sum_{i=1}^k \mathbb{E}_{(X_0, \gamma_0)} [\xi_i]^2. \quad (3.27)$$

To establish (3.21) we need an upper bound on the right hand side of (3.27). An appropriate bound is derived by Łatuszyński et al. [2013a]. Combining (3.12) and (3.14) from the aforementioned paper, we get

$$\mathbb{E}_{(X_0, \gamma_0)} [\xi_i]^2 \leq \sup_{\gamma} \sigma^2(f, P_{\gamma})(n_i + 2\mu),$$

providing an upper bound for every $k \geq 1$,

$$\mathbb{E}_{(X_0, \gamma_0)} \left[\left(\Xi_{N_k}^{(2)} \right)^2 \right] \leq \sup_{\gamma} \sigma^2(f, P_{\gamma})(N_k - 2\mu k). \quad (3.28)$$

Theorems 4.2 and 5.2 of [Latuszyński et al. \[2013a\]](#) and Theorem 6 of Chapter 2 imply that any of the (local) simultaneous drift Assumptions 2, 4, or 3 imply

$$\sup_{\gamma} \sigma^2(f, P_{\gamma}) < \infty.$$

Therefore, together with (3.18) and (3.28), this implies the first part of the proposition, i.e., the MSE convergence (3.21).

We shall now establish the CLT (3.22).

Consider also a filtration $\{\tilde{\mathcal{F}}_n\}$ that is defined as follows. For $(i, j) \in \{(m, 1), \dots, (m, n_m) : m \geq 1\}$, define

$$\tilde{\mathcal{F}}_0 = \{\emptyset\}, \tilde{\mathcal{F}}_{N_{i-1}+j} = \sigma \left\{ \tilde{\mathcal{F}}_{N_{i-1}+j-1} \cup \sigma \left\{ Y_{T_j^{(i)}}^{(i)}, \dots, Y_{T_{j+1}^{(i)}-1}^{(i)} \right\} \cup \{T_{j+1}^{(i)}\} \right\}, \quad (3.29)$$

where $\{Y_n^{(i)}\}$ is defined in (3.19). Let

$$\tilde{\xi}_{i,j}(f) = \xi_{i,j} I_{\{T_j^{(i)} < n_i\}}.$$

Lexicographically ordered sequence $\{\tilde{\xi}_{i,j}\}$ is adapted to the filtration $\{\tilde{\mathcal{F}}_n\}$, i.e., $\xi_{i,j}$ is measurable w.r.t. $\tilde{\mathcal{F}}_{N_{i-1}+j}$. Moreover, since $T_{R_i(n_i)}^{(i)} \leq T_{n_i}^{(i)}$,

$$\xi_i(f) = \sum_{j=1}^{n_i-1} \tilde{\xi}_{i,j}(f),$$

and conditionally on $\tilde{\mathcal{F}}_{N_{i-1}+j-1}$,

$$\mathbb{E}_{(X_0, \gamma_0)} [\tilde{\xi}_{i,j} | \tilde{\mathcal{F}}_{N_{i-1}+j-1}] = I_{\{T_j^{(i)} < n_i\}} \mathbb{E}_{(\nu, \gamma_{N_i})} \left[\xi_{i,j} | \tilde{\mathcal{F}}_{N_{i-1}+j-1} \right]$$

$$= I_{\{T_j^{(i)} < n_i\}} \pi(f) \mu = 0,$$

where the second equality follows from (3.26).

The desired CLT (3.22) would follow from the martingale CLT (see Theorem 2.2 in Dvoretzky [1972]) for $\{\tilde{\xi}_{i,j}(f)\}$, once we show that

$$\frac{1}{N_k} \sum_{i=1}^k \sum_{j=1}^{n_i-1} \mathbb{E}_{(X_0, \gamma_0)} \left[\tilde{\xi}_{i,j}^2 | \tilde{\mathcal{F}}_{N_{i-1}+j-1} \right] \xrightarrow{P} \sigma_\infty^2 \quad (3.30)$$

for $\sigma_\infty^2 > 0$ defined in the statement of the proposition.

Using identity (3.12) of Łatuszyński et al. [2013a], we can write

$$\begin{aligned} & \frac{1}{N_k} \sum_{i=1}^k \sum_{j=1}^{n_i-1} \mathbb{E}_{(X_0, \gamma_0)} \left[\tilde{\xi}_{i,j}^2 | \tilde{\mathcal{F}}_{N_{i-1}+j-1} \right] \\ &= \frac{1}{N_k} \sum_{i=1}^k \sum_{j=1}^{n_i-1} \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \mu I_{\{T_j^{(i)} < n_i\}} \\ &= \frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \mu (R_i(n_i) - 1) \\ &= \frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \mu R_i(n_i) + \mathcal{O} \left(\frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \right) \\ &= \frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) (\mu R_i(n_i) - n_i) + \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \frac{n_i}{N_k} + o(1), \end{aligned}$$

where we used $\mathcal{O} \left(\frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \right) = o(1)$ due to Lemma 6.

Lemma 7. *There exists a constant $M < \infty$ such that*

$$\sup_{\gamma \in \Gamma} \mathbb{E}_{(\nu, \gamma)} \left| \mu R_i(n_i) - n_i \right| \leq M(1 + \sqrt{n_i})$$

It follows from the lemma and (3.18),

$$\frac{1}{N_k} \sum_{i=1}^k \sigma^2 \left(f, P_{\gamma_{N_{i-1}}} \right) \sup_{\gamma \in \Gamma} \mathbb{E}_{(\nu, \gamma)} \left| \mathbb{E}_{(\nu, \gamma)} \left[T^{(i)} \right] R_i(n_i) - n_i \right|$$

$$\leq \sup_{\gamma \in \Gamma} \sigma^2(f, P_\gamma) \sum_{i=1}^k \frac{M(1 + \sqrt{n_i})}{N_k} = \mathcal{O}\left(\frac{k^{1+\beta/2}}{k^{1+\beta}}\right) = \mathcal{O}\left(\frac{1}{k^{\beta/2}}\right).$$

Therefore,

$$\lim_{k \rightarrow \infty} \frac{1}{N_k} \sum_{i=1}^k \sum_{j=1}^{n_i-1} \mathbb{E}_{(X_0, \gamma_0)} \left[\tilde{\xi}_{i,j}^2 | \tilde{\mathcal{F}}_{N_i+j-1} \right] = \lim_{k \rightarrow \infty} \sum_{i=1}^k \sigma^2(f, P_{\gamma_{N_i-1}}) \frac{n_i}{N_k}.$$

Since we assume that $\sigma^2(f, P_\gamma)$ is a continuous function of γ , we have $\sigma^2(f, P_{\gamma_{N_k}}) \rightarrow \sigma_\infty^2$ as $k \rightarrow \infty$, and thus (3.30) holds, whence the CLT (3.22) follows.

□

3.6.1 Proof of Theorem 10

We can control the terms $\Xi_{N_k, N}^{(4)}, \Xi_{N_k}^{(j)}, j \in \{1, 3\}$ from the decomposition (3.20) using the following lemma

Lemma 8. *Under conditions of Theorem 10, there exists $M < \infty$ such that*

$$\sup_j \mathbb{E}_{X_0, \gamma_0} V(X_j) \leq M. \quad (3.31)$$

Jensen's inequality implies

$$\mathbb{E}_{(X_0, \gamma_0)} \left[\left(\Xi_{N_k}^{(1)} \right)^2 \right] \leq k \sum_{i=1}^k \mathbb{E}_{(X_0, \gamma_0)} \left[(\eta_i)^2 \right].$$

Theorem 4.2 of Łatuszyński et al. [2013a] yields that for some $\widehat{M} < \infty$ and all $i \geq 1$,

$$\mathbb{E}_{(X_0, \gamma_0)} [\eta_i^2] \leq \widehat{M} \mathbb{E}_{(X_0, \gamma_0)} [V(X_{N_i})].$$

Thus, together with Lemma 8 we obtain the bound

$$\begin{aligned} & \mathbb{E}_{(X_0, \gamma_0)} \left[\left(\Xi_{N_k}^{(1)} \right)^2 \right] \\ & \leq k^2 \widehat{M} \sup_{j \geq 0} \left(\mathbb{E}_{(X_0, \gamma_0)} [V(X_{N_j})] \right) \leq k^2 \widehat{M} M = \mathcal{O}(k^2). \end{aligned} \quad (3.32)$$

Similarly,

$$\begin{aligned}\mathbb{E}_{(X_0, \gamma_0)} \left[\left(\Xi_{N_k}^{(3)} \right)^2 \right] &\leq k \sum_{i=1}^k \mathbb{E}_{(X_0, \gamma_0)} \left[(\zeta_i)^2 \right] \\ &\leq k^2 \widehat{M} \sup_{j \geq 0} \left(\mathbb{E}_{(X_0, \gamma_0)} [V(X_{N_j})] \right) \leq k^2 \widehat{M} M = \mathcal{O}(k^2).\end{aligned}\tag{3.33}$$

Using the decomposition (3.20) and bounds (3.21), (3.32) and (3.33), the triangle inequality yields

$$E_{(X_0, \gamma_0)} \left[S_{N_k}(f) \right]^2 = \mathcal{O}(N_k) + \mathcal{O}(k^2) + \mathcal{O}(k^2).\tag{3.34}$$

Notice, the adaptive chain $\{X_n\}$ is Markov on the interval $[N_k, N]$ and thus, Theorem 4.2 of [Latuszyński et al. \[2013a\]](#) can be applied to bound $\mathbb{E}_{X_0, \gamma_0} \left[\Xi_{N_k, N}^{(4)} \right]^2$. We get, that for some $\widehat{M} < \infty$,

$$\mathbb{E}_{X_0, \gamma_0} \left[\Xi_{N_k, N}^{(4)} \right]^2 \leq \widehat{M} n_k \sup_{j \geq 0} \left(\mathbb{E}_{(X_0, \gamma_0)} [V(X_{N_j})] \right) = \mathcal{O}(n_k) = \mathcal{O}(k^\beta),\tag{3.35}$$

where we used the theorem assumption $n_k = \mathcal{O}(k^\beta)$.

Finally, (3.34) and (3.35) combined together imply

$$MSE(\hat{\pi}_N(f)) = \mathcal{O}\left(\frac{1}{N_k}\right) + \mathcal{O}\left(\frac{k^2}{N_k^2}\right) = \mathcal{O}\left(\frac{1}{k^{1+\beta}}\right) + \mathcal{O}\left(\frac{1}{k^{2\beta}}\right),\tag{3.36}$$

where for the second equality we used (3.18).

We shall prove every statement of the theorem below.

i) If $\beta \in [0, 1]$, the right hand side of (3.36) converges to zero at rate $k^{2\beta}$, which is due to (3.18) equal to the rate of $N^{\frac{2\beta}{1+\beta}}$.

ii) If $\beta \geq 1$, the rate of convergence in (3.36) is $k^{1+\beta}$, which is due to (3.18) precisely the rate at which N grows.

iii) For $\beta > 1/2$ we have that for any $\varepsilon > 0$, using (3.36) and Chebyshev's inequality,

$$\mathbb{P}_{(X_0, \gamma_0)} (|\hat{\pi}_{N_k}(f)| > \varepsilon) = \mathcal{O}\left(\frac{1}{k^{1+\beta}}\right) + \mathcal{O}\left(\frac{1}{k^{2\beta}}\right),$$

so that

$$\sum_{k \geq 1} \mathbb{P}_{(X_0, \gamma_0)} (|\hat{\pi}_{N_k}(f)| > \varepsilon) < \infty$$

and by Borel-Cantelli lemma we ensure that $\limsup_{k \rightarrow \infty} |\hat{\pi}_{N_k}(f)| < \varepsilon$. Since

$$\hat{\pi}_N = \frac{N_k}{N} \hat{\pi}_{N_k} + \frac{1}{N} \Xi_{N_k, N}^{(4)},$$

in order to get the SLLN for $\hat{\pi}_N$, it is enough to show that $\frac{1}{N} \Xi_{N_k, N}^{(4)} \xrightarrow{a.s.} 0$. Chebyshev's inequality and (3.35) imply that for some $M < \infty$

$$\sum_{N \geq 1} \mathbb{P}_{(X_0, \gamma_0)} \left(\left| \Xi_{N_k, N}^{(4)} \right| \geq N\varepsilon \right) \leq M \sum_{k \geq 1} \frac{n_k^2}{N_k^2}, \quad (3.37)$$

where we used $N \geq N_k$. (3.14) and (3.18) imply that $\frac{n_k^2}{N_k^2} = \mathcal{O}\left(\frac{1}{k^2}\right)$ so that the right hand side of (3.37) is finite, whence using Borel-Cantelli lemma, we conclude the SLLN for $\hat{\pi}_N$.

iv) We shall use Proposition 10. In order to get the CLT for (3.22) we need to show continuity of the asymptotic variance $\sigma^2(f, P_\gamma)$ in $\gamma \in \Gamma$ for functions f such that $\|f\|_{V^{1/2}} < \infty$.

Recall that without loss of generality we assume $\pi(f) = 0$. From Section 17.4.2 of [Meyn and Tweedie \[2009\]](#), the asymptotic variance in the CLT can be written as

$$\sigma^2(f, P_\gamma) = \pi(\hat{f}^2 - \{P_\gamma(\hat{f})\}^2) = 2\pi(\hat{f}f) - \pi(f^2), \quad (3.38)$$

where $\hat{f} = \hat{f}^{(\gamma)}$ solves the Poisson equation

$$\hat{f} - P_\gamma(\hat{f}) = f. \quad (3.39)$$

For parameters $\gamma_1, \gamma_2 \in \Gamma$, we can bound

$$\left| \sigma^2(f, P_{\gamma_1}) - \sigma^2(f, P_{\gamma_2}) \right| \leq 2\pi \left(|\hat{f}^{(\gamma_1)} - \hat{f}^{(\gamma_2)}| \cdot f \right) \quad (3.40)$$

$$\leq 2M\pi \left(|\hat{f}^{(\gamma_1)} - \hat{f}^{(\gamma_2)}| \cdot V^{1/2} \right) \leq M\pi(V) \|\hat{f}^{(\gamma_1)} - \hat{f}^{(\gamma_2)}\|_{V^{1/2}}, \quad (3.41)$$

where we used that for some $M < \infty$,

$$|f| \leq MV^{1/2}$$

and

$$|\hat{f}^{(\gamma_1)} - \hat{f}^{(\gamma_2)}| \leq V^{1/2} \|\hat{f}^{(\gamma_1)} - \hat{f}^{(\gamma_2)}\|_{V^{1/2}}.$$

Under conditions of the theorem it follows from Section 4.2 of [Glynn and Meyn \[1996\]](#) that $\|\hat{f}^{(\gamma)}\|_{V^{1/2}} < \infty$ and $\hat{f}^{(\gamma)}$ is continuous in $V^{1/2}$ -norm as a function of γ . Combining these observations together with (3.40), we conclude that $\sigma_\gamma^2(f)$ is a continuous function of γ , so that

$$\sigma^2(f, P_{\gamma_{N_{i-1}}}) \rightarrow \sigma_\infty^2 := \sigma^2(f, P_{\gamma_\infty}),$$

whence (3.22) follows.

It is left to notice that (3.18), (3.32), (3.33) and (3.35) imply that $\frac{1}{\sqrt{N}}\Xi_{N_k, N}^{(4)} \xrightarrow{P} 0$ and $\frac{1}{\sqrt{N}}\Xi_N^{(i)} \xrightarrow{P} 0$ for $i \in \{1, 3\}$, if $\beta > 1$.

□

3.6.2 Proof of Theorem 11

Let V_1, \dots, V_m and F_1, \dots, F_m be the finite collection of drift functions and finite partition of Γ from Theorem 6 of Chapter 2. On Γ define a function r that maps $r(\gamma) = j$ if $\gamma \in F_j$. Theorem 7 of Chapter 2 implies that

$$\sup_n \mathbb{E}_{(X_0, \gamma_0)} V_{r(\gamma_n)}(X_n) < \infty.$$

The rest of the proof is identical to the proof of Theorem 10 where $V(x)$ is substituted with $V_{r(\gamma)}(x)$ and $V(X_n)$ with $V_{r(\gamma_n)}(X_n)$.

□

3.6.3 Proof of Theorem 12

In view of Proposition 10, (3.21) together with the Chebyshev's inequality imply that for any $\varepsilon > 0$,

$$\mathbb{P}_{(X_0, \gamma_0)} \left(\left| \frac{\Xi_{N_k}^{(2)}}{N_k} \right| \geq \varepsilon \right) = \mathcal{O} \left(\frac{1}{N_k} \right). \quad (3.42)$$

Lemma 8 that we used to control $\Xi_{N_k}^{(1)}$, $\Xi_{N_k}^{(3)}$, $\Xi_{N_k, N}^{(4)}$ in the proof of Theorem 10 does not apply for the polynomial ergodicity Assumption 3. On the other hand, the following alternative holds.

Lemma 9. *Under conditions of Theorem 12, there exists $M < \infty$ such that for all $n > 0$ and $m \geq M$,*

$$\mathbb{P}_{(X_0, \gamma_0)} (V^{2\alpha-1}(X_n) > m) \leq MV(X_0) \frac{\log(1+m)}{m}. \quad (3.43)$$

Lemma 9 implies that for arbitrary fixed $\delta > 0$,

$$\sum_{i=k}^{\infty} \mathbb{P}_{(X_0, \gamma_0)} (V^{2\alpha-1}(X_{N_k}) > k^{1+\delta}) < \infty. \quad (3.44)$$

Define sets.

$$E_k := \left\{ V(X_{N_k}) < k^{\frac{1+\delta}{2\alpha-1}} \right\}, \quad A_m = \cap_{k \geq m} E_k. \quad (3.45)$$

Borel-Cantelli lemma together with (3.44) imply

$$\mathbb{P}_{(X_0, \gamma_0)} \left(\liminf_{k \rightarrow \infty} E_k \right) = 1. \quad (3.46)$$

and, in particular, for every $m \geq 1$ we have

$$\lim_{m \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} (A_m) = 1. \quad (3.47)$$

Lemma 9 and (3.47) imply that for every $\varepsilon > 0$, $m \geq 1$ and $s > 0$,

$$\lim_{k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(A_m, \left| \frac{\Xi_{N_k}^{(1)}}{N_k^s} \right| > \varepsilon \right) = \lim_{k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(\frac{1}{N_k^s} \left| \sum_{i=1}^k \eta_i I_{E_i} \right| > \varepsilon \right), \quad (3.48)$$

$$\lim_{k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(A_m, \left| \frac{\Xi_{N_k}^{(3)}}{N_k^s} \right| > \varepsilon \right) = \lim_{N_k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(\frac{1}{N_k^s} \left| \sum_{i=1}^k \zeta_i I_{E_i} \right| > \varepsilon \right), \quad (3.49)$$

where we notice,

$$\begin{aligned} \mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k}^{(1)}}{N_k^s} \right| > \varepsilon \right) &= \mathbb{P}_{(X_0, \gamma_0)} \left(\frac{1}{N_k^s} \left| \sum_{i=1}^k \eta_i I_{E_i} \right| > \varepsilon \right), \\ \mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k}^{(3)}}{N_k^s} \right| > \varepsilon \right) &= \mathbb{P}_{(X_0, \gamma_0)} \left(\frac{1}{N_k^s} \left| \sum_{i=1}^k \zeta_i I_{E_i} \right| > \varepsilon \right). \end{aligned}$$

Jensen's inequality, Theorem 5.2 of [Łatuszyński et al. \[2013b\]](#), (3.45) and Lemma 6 imply that for some $\widehat{M} < \infty$,

$$\begin{aligned} \mathbb{E}_{(X_0, \gamma_0)} \left[\left(\sum_{i=1}^k \eta_i I_{E_i} \right)^2 \right] &\leq k \sum_{i=1}^k \mathbb{E}_{X_0, \gamma_0} [\eta_i I_{E_i}]^2 \leq \\ &\leq k \widehat{M} \sum_{i=1}^k \mathbb{E} \left[V^\alpha(X_{N_i}) I_{E_i} \right] \leq k \widehat{M} \sum_{i=1}^k i^{\frac{\alpha(1+\delta)}{2\alpha-1}} = \mathcal{O} \left(k^{2+\frac{\alpha(1+\delta)}{2\alpha-1}} \right), \end{aligned}$$

and, similarly,

$$\mathbb{E}_{(X_0, \gamma_0)} \left[\left(\sum_{i=1}^k \zeta_i I_{E_i} \right)^2 \right]^2 = \mathcal{O} \left(k^{2+\frac{\alpha(1+\delta)}{2\alpha-1}} \right)$$

Applying Chebyshev's inequality to (3.48) and (3.49) we obtain,

$$\mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k}^{(1)}}{N_k^s} \right| > \varepsilon \right) = \mathcal{O} \left(\frac{k^{2+\frac{\alpha(1+\delta)}{2\alpha-1}}}{N_k^{2s}} \right) = \mathcal{O} \left(k^{\frac{\alpha(1+\delta)}{2\alpha-1} + (2-2s) - 2\beta s} \right), \quad (3.50)$$

$$\mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k}^{(3)}}{N_k^s} \right| > \varepsilon \right) = \mathcal{O} \left(\frac{k^{2+\frac{\alpha(1+\delta)}{2\alpha-1}}}{N_k^{2s}} \right) = \mathcal{O} \left(k^{\frac{\alpha(1+\delta)}{2\alpha-1} + (2-2s) - 2\beta s} \right), \quad (3.51)$$

where we used (3.18).

Since the adaptive chain $\{X_n\}$ is Markov on $[N_k, N]$, we can apply Theorem 5.2 of [Łatuszyński et al. \[2013a\]](#) to bound $\Xi_{N_k, N}^{(4)}$:

$$\mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k, N}^{(4)}}{N_k^s} \right| > \varepsilon \right) = \mathcal{O} \left(\frac{k^{\frac{\alpha(1+\delta)}{2\alpha-1}} n_k}{N_k^{2s}} \right) = \mathcal{O} \left(\frac{k^{\frac{\alpha(1+\delta)}{2\alpha-1} + \beta}}{k^{2s+2\beta s}} \right) \quad (3.52)$$

and for all $m \geq 1$,

$$\lim_{k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(A_m, \left| \frac{\Xi_{N_k, N}^{(4)}}{N_k^s} \right| > \varepsilon \right) = \lim_{k \rightarrow \infty} \mathbb{P}_{(X_0, \gamma_0)} \left(A_1, \left| \frac{\Xi_{N_k, N}^{(4)}}{N_k^s} \right| > \varepsilon \right). \quad (3.53)$$

We shall prove every statement of the theorem below.

i) If $\beta > \frac{\alpha}{4\alpha-2}$, then for sufficiently small $\delta > 0$,

$$\lim_{N \rightarrow \infty} k^{\frac{\alpha(1+\delta)}{2\alpha-1} - 2\beta} = 0.$$

Therefore, the right hand side of (3.42), (3.50), (3.51) and (3.52) converges to zero when $s = 1$. Therefore, by taking limit $m \rightarrow \infty$ in (3.48), (3.49) and (3.53) we derive the WLLN for $\hat{\pi}_N$.

ii) For $\beta > 1/2 + \frac{\alpha}{4\alpha-2}$, in the same manner as in the proof of Theorem 10, using (3.42) and (3.50), (3.51), we could establish that

$$\sum_{k \geq 1} \mathbb{P}_{(X_0, \gamma_0)} (A_1, |\hat{\pi}_{N_k}(f)| > \varepsilon) < \infty.$$

and use Borel-Cantelli lemma to establish the SLLN for $\hat{\pi}_{N_k}(f)I_{\{A_1\}}$. Then from (3.52) and Borel-Cantelli lemma, we could derive the SLLN for $\hat{\pi}_N(f)I_{\{A_1\}}$ and use (3.46) to ensure that the SLLN holds for $\hat{\pi}_N(f)$.

iii) We shall use Proposition 10 in order to get the CLT for $\frac{\Xi_{N_k}^{(2)}}{\sqrt{N_k}}$. The CLT would follow if we show that $\sigma^2(f, P_\gamma)$ is a continuous function of γ .

Consider the following representation of the asymptotic variance (see, e.g., Section 17.4.3 of [Meyn and Tweedie \[2009\]](#)):

$$\sigma^2(f, P_\gamma) = \pi(f^2) + 2 \sum_{i=1}^{\infty} \mathbb{E}_{(\pi, \gamma)} f(X_0) f(X_i),$$

where without loss of generality we assume $\pi(f) = 0$.

It is known that $\|P_\gamma^n - \pi\|_{V^{\alpha 3/2-1}}$ converges to zero at a polynomial rate (see, e.g., 3.6 of [Jarner and Roberts \[2002\]](#)). Theorem 6 of [Fort and Moulines \[2003\]](#) provides a quantitative bound on the rate of convergence in terms of polynomial drift coefficients. In particular, it follows that for any $\kappa \in \left[1, \frac{1}{1-\alpha}\right]$ and $\delta > 0$, there exists some $M = M(\kappa) < \infty$, such that

$$n^{\kappa-1-\delta} \|P_\gamma^n(x, \cdot) - \pi(\cdot)\|_{V^{1-\kappa(1-\alpha)}} \leq M V^{1-\kappa(1-\alpha)}(x).$$

By the theorem assumption $\alpha > 2/3$. Thus, for $\kappa = \frac{2-\alpha 3/2}{1-\alpha}$ and appropriate $\delta > 0$, we have

$$n^{3/2} \|P_\gamma^n(x, \cdot) - \pi(\cdot)\|_{V^{\alpha 3/2-1}} \leq M V^{\alpha 3/2-1}(x).$$

Note that

$$\mathbb{E}_{(x,\gamma)} f(X_0) f(X_i) < |f(x)| \|P_\gamma^i(x, \cdot) - \pi(\cdot)\|_{V^{\alpha 3/2-1}} \leq M i^{-3/2} V^{3\alpha-2}(x).$$

Since $\pi(V^{3\alpha-2}) < \infty$ (see Proposition 5.4 of [Latuszyński et al. \[2013a\]](#)), we have that for any $\varepsilon > 0$, there exists $N = N(\varepsilon) < \infty$, such that

$$\sigma^2(f, P_\gamma) \leq \pi(f^2) + 2 \sum_{i=1}^N \mathbb{E}_{(\pi, \gamma)} f(X_0) f(X_i) + \varepsilon. \quad (3.54)$$

For any parameters $\gamma \in \Gamma$ and a sequence $\{\gamma_n\} \subset \Gamma$, (3.54) implies

$$\begin{aligned} \left| \sigma^2(f, P_\gamma) - \sigma^2(f, P_{\gamma_n}) \right| &= 2 \left| \sum_{i=1}^N \mathbb{E}_{(\pi, \gamma)} f(X_0) f(X_i) \right. \\ &\quad \left. - \sum_{i=1}^N \mathbb{E}_{(\pi, \gamma_n)} f(X_0) f(X_i) \right| + \varepsilon \\ &\leq 2 \sum_{i=1}^N \int |f(y)| \left| (P_\gamma^i f)(y) - (P_{\gamma_n}^i f)(y) \right| \pi(dy) + \varepsilon. \end{aligned} \quad (3.55)$$

Since P_γ is a continuous operator in $V^{\alpha 3/2-1}$ -norm, there exists $\tilde{\delta} > 0$, such that for $\|\gamma - \gamma_n\| < \tilde{\delta}$, and $i \in \{1, \dots, N\}$,

$$\sup_x \frac{\|P_\gamma^i(x, \cdot) - P_{\gamma_n}^i(x, \cdot)\|_{V^{\alpha 3/2-1}}}{V^{\alpha 3/2-1}(x)} \leq \frac{\varepsilon}{N \pi(V^{3\alpha-2})},$$

where we note that $\pi(V^{3\alpha-2}) < \infty$ (see Proposition 5.4 of [Latuszyński et al. \[2013a\]](#)).

Therefore, since $|f| \leq \widehat{M} V^{\alpha 3/2-1}$ for some $\widehat{M} < \infty$, (3.55) implies

$$\left| \sigma^2(f, P_\gamma) - \sigma^2(f, P_{\gamma_n}) \right| \leq 2 \widehat{M}^2 \sum_{i=1}^N \int \frac{\varepsilon}{N \pi(V^{3\alpha-2})} V^{3\alpha-2} \pi(dy) + \varepsilon$$

$$= \left(\widehat{M}^2 + 1 \right) \varepsilon.$$

We conclude that $\sigma^2(f, P_\gamma)$ is a continuous function of γ . Thus, (3.22) follows.

Taking $s = 1/2$ in (3.48) - (3.53), we conclude that for $\beta > 1 + \frac{\alpha}{2\alpha-1}$, we have $\frac{1}{\sqrt{N}} \Xi_{N_k, N}^{(4)} \xrightarrow{P} 0$ and $\frac{1}{\sqrt{N}} \Xi_N^{(i)} \xrightarrow{P} 0$ for $i \in \{1, 3\}$.

□

Proof of Proposition 5.

Lemma 10. *For any $\delta > 0$ and $p > 2 + \delta$,*

$$\begin{aligned} & \mathbb{E}_{(\nu, \gamma)} \left[\sum_{j=0}^{T-1} f(X_j) \right]^{2+\delta} \\ & \leq \left(\mathbb{E}_{(\nu, \gamma)} \left[T^{\frac{(2+\delta)(p-1)}{p-2-\delta}} \right] \right)^{\frac{p-2-\delta}{p}} \left(\mathbb{E}_{(\nu, \gamma)} \left[\sum_{j=0}^{T-1} f(X_j)^p \right] \right)^{\frac{2+\delta}{p}}. \end{aligned} \quad (3.56)$$

From Theorem 4.1 of [Roberts and Tweedie \[1999\]](#) it follows that for any $\kappa > 1$, there exists a constant $C(\kappa)$ depending only on the drift coefficients, such that

$$\mathbb{E}_{(\nu, \gamma)} [T^\kappa] \leq C(\kappa),$$

implying that

$$\sup_{\gamma} \mathbb{E}_{(\nu, \gamma)} [T^\kappa] < \infty.$$

We are left to show that we can find $p > 2$, such that

$$\sup_{\gamma \in \Gamma} \mathbb{E}_{(\nu, \gamma)} \left[\sum_{j=0}^{T-1} f(X_j)^p \right] < \infty \quad (3.57)$$

By the assumption of the proposition, the function f is such that $\|f\|_{V^{1/2-\delta}} < \infty$ for some $\delta > 0$. Therefore, there exists $p > 2$, such that $|f^p(x)| \leq MV(x)$ for some $M < \infty$ and all x . Identity (3.26) yields (3.57), which finishes the proof.

□

Proof of Proposition 6.

Let V_1, \dots, V_m be the finite collection of drift functions from Theorem 6 of Chapter 2. As in the proof of Proposition 5, we can use Theorem 4.1 of [Roberts and Tweedie \[1999\]](#), to establish that for any $\kappa > 1$ there exists a constant $C(\kappa)$ depending only on the drift coefficients such that $\mathbb{E}_{(\nu, \gamma)} [T^\kappa] \leq C(\kappa)$, so that $\sup_{\gamma} \mathbb{E}_{(\nu, \gamma)} [T^\kappa] < \infty$.

∞ , and thus, conclude the proposition statement.

□

Proof of Proposition 7.

From Theorem 4 of Douc et al. [2008] it follows that there exists a constant C depending only on the drift coefficients such that

$$\mathbb{E}_{(\nu, \gamma)} \left[T^{\frac{\alpha}{1-\alpha}} \right] \leq C,$$

implying that

$$\sup_{\gamma} \mathbb{E}_{(\nu, \gamma)} \left[T^{\frac{\alpha}{1-\alpha}} \right] < \infty. \quad (3.58)$$

We shall use Lemma 10. For the right hand side of (3.56) to be finite for some $\delta > 0$, we need:

- (a) $\|f^p\|_{V^\alpha} < \infty$ (see Proposition 5.4 of Łatuszyński et al. [2013a]);
- (b) $\mathbb{E}_{(\nu, \gamma)} \left[T^{\frac{(2+\delta)(p-1)}{p-2-\delta}} \right] < \infty$ for some $\delta < 0$.

It follows from (3.58), that in order to satisfy (b), α and p should be chosen so that

$$\frac{2(p-1)}{p-2} < \frac{\alpha}{1-\alpha}.$$

Since $p > 2$ and $\alpha > 2/3$, we have to choose p such that

$$p > \frac{4\alpha - 2}{3\alpha - 2}.$$

Note that $\|f^p\|_{V^\alpha} < \infty$ iff $\|f\|_{V^{\alpha/p}} < \infty$. Thus, we conclude that any function f for which $\|f\|_{V^{\frac{\alpha(3\alpha-2)}{4\alpha-2}-\delta}} < \infty$ for some $\delta > 0$, satisfies (a) and (b), and thus, the Assumption 5 holds for f .

□

3.6.4 Proof of Theorem 13

Ergodicity follows from Theorem 3 of Bai et al. [2011] in case conditions (a) holds, and from Theorem 7 of Chapter 2 in case conditions (b) are satisfied.

For the case (c), we could use Theorem 5 of Bai et al. [Bai et al. \[2011\]](#), provided that there exists $b' > b$ such that for all $x \notin C$

$$cV^\alpha(x) \geq b'. \quad (3.59)$$

However, since we assume that all level sets of V are uniform small sets, the condition (3.59) is fulfilled by virtue of Corollary A.2 of [Atchadé and Fort \[2010\]](#).

□

3.6.5 Proof of Theorem 14

Since sequence $\{n_i\}$ satisfies (3.14), in order to prove statements of Theorems 10, 11, or 12 we could literally repeat the proofs of the theorems, where the filtrations (3.24) and (3.29) should be substituted with

$$\mathcal{F}_0 = \{\emptyset\}, \mathcal{F}_i = \sigma\left\{\mathcal{F}_{i-1} \cup \{Y_0^{(i)}, \dots, Y_{T_{R(n_i)}^{(i)}-1}^{(i)}\} \cup \{n_i\}\right\},$$

and for $(i, j) \in \{(m, 1), \dots, (m, n_m^* + \lfloor n_m^* \rfloor^\delta) : m \geq 1\}$, with

$$\tilde{\mathcal{F}}_0 = \{\emptyset\}, \tilde{\mathcal{F}}_{N_{i-1}^*+j} = \sigma\left\{\tilde{\mathcal{F}}_{N_{i-1}^*+j-1} \cup \sigma\left\{Y_{T_j^{(i)}}^{(i)}, \dots, Y_{T_{j+1}^{(i)}-1}^{(i)}\right\} \cup \{T_{j+1}^{(i)}\} \cup \{n_i\}\right\},$$

respectively. Here we set $N_k^* = \sum_{i=0}^k (n_i^* + \lfloor n_i^* \rfloor^\delta)$ and $n_0^* = 0$.

It is left to notice that the diminishing condition (3.15) holds, since kernels P_{γ_n} and $P_{\gamma_{n+1}}$ are the same with high probability by construction of Algorithm 18.

□

3.7 Appendix A

Proof of Proposition 8. One can easily see that $C := \{1, 3\}$ is a small set for P_γ , $\gamma \in \Gamma$, i.e., (3.8) holds. Also define a function V as: $V(1) = V(3) = 1$, $V(2) = V(4) = 8$; constant $\lambda := \frac{7}{8}$. Then for any ε such that $1 - \varepsilon - \varepsilon^3 \geq 2\varepsilon$, the simultaneous geometric drift condition (3.9) holds. Indeed,

$$\begin{aligned} P_1 V(2) &= \frac{1}{2} \times V(1) + \frac{1}{2} \times V(3) = 1 < 7 = \lambda V(2), \\ P_2 V(2) &= \frac{1}{4} \times V(1) + \frac{1}{4} \times V(3) + \frac{1}{4} \times V(4) + \frac{1}{4} \times V(2) = \frac{9}{2} < 7 = \lambda V(2), \end{aligned}$$

and

$$\begin{aligned}
P_1V(4) &= \frac{1}{2} \times V(3) + \frac{1}{2} \times V(4) = \frac{9}{2} < 7 = \lambda V(4), \\
P_2V(4) &= \frac{1}{4} \times V(3) + \frac{1}{4} \times V(2) \times \frac{2\varepsilon^3}{1-\varepsilon-\varepsilon^3} + \\
&+ \frac{1}{4} \times V(4) \times \left(1 - \frac{2\varepsilon^3}{1-\varepsilon-\varepsilon^3}\right) + \frac{1}{2} \times V(4) = \frac{25}{4} < 7 = \lambda V(4).
\end{aligned}$$

Therefore, by virtue of Theorem 10, the SLLN holds.

However, the adaptive chain fails to be ergodic for small enough $\varepsilon > 0$ (recall that $\pi(1) = \varepsilon$). It suffices to show that for some $\delta > 0$ and small enough $\varepsilon > 0$,

$$\limsup_{k \rightarrow \infty} \mathbb{P}(X_{2^{k^2}+2} = 1) > \pi(1) + \delta. \quad (3.60)$$

Using Markov property and the definition of the Algorithm 19, we get,

$$\begin{aligned}
&\mathbb{P}(X_{2^{k^2}+2} = 1 | \gamma_{2^{k^2}} = 1) \\
&\geq \mathbb{P}(X_{2^{k^2}+2} = 1, X_{2^{k^2}+1} = 3, X_{2^{k^2}} = 4, | \gamma_{2^{k^2}} = 1) \\
&+ \mathbb{P}(X_{2^{k^2}+2} = 1, X_{2^{k^2}+2} = 1, X_{2^{k^2}} = 1, | \gamma_{2^{k^2}} = 1) \\
&= P_2(X_{2^{k^2}+2} = 1 | X_{2^{k^2}+1} = 3) \times P_1(X_{2^{k^2}+1} = 3 | X_{2^{k^2}} = 4) \\
&\times \mathbb{P}(X_{2^{k^2}+2} = 4 | \gamma_{2^{k^2}} = 1) \\
&+ P_1(X_{2^{k^2}+2} = 1 | X_{2^{k^2}+1} = 1) \times P_2(X_{2^{k^2}+1} = 1 | X_{2^{k^2}} = 1) \\
&\times \mathbb{P}(X_{2^{k^2}} = 1 | \gamma_{2^{k^2}} = j) \\
&= \frac{1}{4} \frac{2\varepsilon}{1-\varepsilon-\varepsilon^3} \times \frac{1}{2} \times \mathbb{P}(X_{2^{k^2}} = 4 | \gamma_{2^{k^2}} = 1) \\
&+ \left(\frac{1}{2} + \frac{1}{2}(1-\varepsilon^2)\right) \times \left(\frac{1}{2} + \frac{1}{2}(1-\varepsilon^2)\right) \times \mathbb{P}(X_{2^{k^2}} = 1 | \gamma_{2^{k^2}} = j).
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\mathbb{P}(X_{2^{k^2}+2} = 1 | \gamma_{2^{k^2}} = 2) \\
&\geq \mathbb{P}(X_{2^{k^2}+2} = 1, X_{2^{k^2}+1} = 3, X_{2^{k^2}} = 4, | \gamma_{2^{k^2}} = 2) \\
&+ \mathbb{P}(X_{2^{k^2}+2} = 1, X_{2^{k^2}+2} = 1, X_{2^{k^2}} = 1, | \gamma_{2^{k^2}} = 2) \\
&+ \mathbb{P}(X_{2^{k^2}+2} = 1, X_{2^{k^2}+2} = 1, X_{2^{k^2}} = 3, | \gamma_{2^{k^2}} = 2) \\
&= P_2(X_{2^{k^2}+2} = 1 | X_{2^{k^2}+1} = 3) \times P_2(X_{2^{k^2}+1} = 3 | X_{2^{k^2}} = 4) \\
&\times \mathbb{P}(X_{2^{k^2}+2} = 4 | \gamma_{2^{k^2}} = 2) \\
&+ P_1(X_{2^{k^2}+2} = 1 | X_{2^{k^2}+1} = 1) \times P_2(X_{2^{k^2}+1} = 1 | X_{2^{k^2}} = 1)
\end{aligned}$$

$$\begin{aligned}
& \times \mathbb{P}(X_{2^{k^2}} = 1 | \gamma_{2^{k^2}} = 2) \\
& + P_1(X_{2^{k^2}+2} = 1 | X_{2^{k^2}+1} = 1) \times P_2(X_{2^{k^2}+1} = 1 | X_{2^{k^2}} = 3) \\
& \times \mathbb{P}(X_{2^{k^2}} = 3 | \gamma_{2^{k^2}} = 2) \\
& = \frac{1}{4} \frac{2\varepsilon}{1 - \varepsilon - \varepsilon^3} \times \frac{1}{4} \times \mathbb{P}(X_{2^{k^2}} = 4 | \gamma_{2^{k^2}} = 1) \\
& + \left(\frac{1}{2} + \frac{1}{2}(1 - \varepsilon^2) \right) \times \left(\frac{1}{2} + \frac{1}{4}(1 - \varepsilon^2) \right) \times \mathbb{P}(X_{2^{k^2}} = 1 | \gamma_{2^{k^2}} = j) \\
& + \left(\frac{1}{2} + \frac{1}{2}(1 - \varepsilon^2) \right) \times \frac{1}{4} \frac{2\varepsilon}{1 - \varepsilon - \varepsilon^3} \times \mathbb{P}(X_{2^{k^2}} = 3 | \gamma_{2^{k^2}} = 2).
\end{aligned}$$

For $j \in \{1, 2\}$,

$$\lim_{k \rightarrow \infty} \mathbb{P}(X_{2^{k^2}} = 1 | \gamma_{2^{k^2}} = j) = \pi(1) = \varepsilon,$$

$$\lim_{k \rightarrow \infty} \mathbb{P}(X_{2^{k^2}} = 3 | \gamma_{2^{k^2}} = j) = \pi(3) = \frac{1 - \varepsilon - \varepsilon^3}{2},$$

and

$$\lim_{k \rightarrow \infty} \mathbb{P}(X_{2^{k^2}} = 4 | \gamma_{2^{k^2}} = j) = \pi(4) = \frac{1 - \varepsilon - \varepsilon^3}{2},$$

whence (3.60) follows, which finishes the proof.

□

Proof of Proposition 9. For every $\gamma := (Z, \nu)$ let P_γ be the Metropolis-Hastings kernel corresponding to the proposal Q_γ . Let the corresponding acceptance ratio be $\alpha_\gamma(x, y) = \min \left\{ 1, \frac{\pi(y)q_\gamma(y, x)}{\pi(x)q_\gamma(x, y)} \right\}$, where $q_\gamma(y, x)$ is the density of Q_γ w.r.t. the Lebesgue measure.

Let P_0 be the Metropolis-Hastings kernel that corresponds to a proposal $Q_0(x, \cdot) = N(x, \kappa I)$ with the corresponding density $q_0(x, y)$. One can conclude from the representation (3.17) that for the Gaussian and Matérn kernels there exists $\beta > 0$, such that for a matrix norm $\|\cdot\|$,

$$\|M(Z, x)\| = \mathcal{O} \left(\exp \left(- \max_{i \in \{1, \dots, t\}} |Z_i - x| / \beta \right) \right), \quad |x| \rightarrow \infty, \quad (3.61)$$

where we used an asymptotic result for modified Bessel functions

$P_v(|x|) \sim \sqrt{\pi/2|x|} \exp(-|x|)$, $|x| \rightarrow \infty$ (see equation 10.25.3 DLMF).

Since the target distribution π has super-exponential tails and satisfies the curvature condition, it follows from Theorem 4.3 of [Järner and Hansen \[2000\]](#), that the kernel P_0 is geometrically ergodic, in particular, the drift function can be chosen as $V(x) := \frac{a}{\sqrt{\pi(x)}} \geq 1$ for some constant $0 < a < \infty$, so that

$$\limsup_{|x| \rightarrow \infty} \frac{P_0 V(x)}{V(x)} < 1.$$

We shall show that (3.61) implies that for any bounded closed (i.e., compact) set Γ

$$\limsup_{|x| \rightarrow \infty} \sup_{\gamma \in \Gamma} \frac{P_\gamma V(x)}{V(x)} < 1, \quad (3.62)$$

whence we conclude that Assumption 2 holds. Note that, it is easy to check that the simultaneous minorisation Assumption 1 holds, since $\kappa > 0$ in the definition of Q_γ , (3.16).

We observe that (3.62) follows if we show that for every $\varepsilon > 0$, there exists $T < \infty$, such that

$$\sup_{\gamma \in \Gamma, |x| > T} \frac{|P_\gamma V(x) - P_0 V(x)|}{V(x)} < \varepsilon. \quad (3.63)$$

One can rewrite the difference

$$\begin{aligned} P_\gamma V(x) - P_0 V(x) &= \int V(y) \alpha_\gamma(x, y) q_\gamma(x, y) dy - \int V(y) \alpha_0(x, y) q_0(x, y) dy \\ &+ V(x) \int (\alpha_0(x, y) q_0(x, y) - \alpha_\gamma(x, y) q_\gamma(x, y)) dy, \end{aligned}$$

where $\alpha_0(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\}$. Since (3.61) holds,

$$\limsup_{|x| \rightarrow \infty} \int |\alpha_\gamma(x, y) q_\gamma(x, y) - \alpha_0(x, y) q_0(x, y)| dy = 0.$$

Therefore, to establish (3.63), it suffices to show that for large T ,

$$\sup_{\gamma \in \Gamma, |x| > T} \frac{1}{V(x)} \int V(y) \left| \alpha_\gamma(x, y) q_\gamma(x, y) dy - \alpha_0(x, y) q_0(x, y) \right| dy < \varepsilon.$$

Let $h_\gamma(x, y) = V(y) \left| \alpha_\gamma(x, y) q_\gamma(x, y) - \alpha_0(x, y) q_0(x, y) \right|$ and $I_\gamma(x) = \int h_\gamma(x, y) dy$. Introduce sets

$$A_1 = A_1(x) = \{y : \pi(y) > \pi(x)\},$$

$$A_2 = A_2(x) = \left\{ y : \frac{\pi(y)}{\pi(x)} \frac{q_\gamma(y, x)}{q_\gamma(x, y)} > 1 \right\},$$

and rewrite

$$I_\gamma(x) = \int_{A_1^c \cap A_2^c} h_\gamma(x, y) dy + \int_{A_1 \cap A_2} h_\gamma(x, y) dy + \int_{A_1 \cap A_2^c} h_\gamma(x, y) dy \quad (3.64)$$

$$+ \int_{A_1^c \cap A_2} h_\gamma(x, y) dy =: I_1(x, \gamma) + I_2(x, \gamma) + I_3(x, \gamma) + I_4(x, \gamma). \quad (3.65)$$

We obtain the following bounds.

$$\begin{aligned} \frac{I_1(x, \gamma)}{V(x)} &= \int_{A_1^c \cap A_2^c} |q_\gamma(y, x) - q_0(x, y)| \frac{\pi(y)}{\pi(x)} \frac{V(y)}{V(x)} dy \\ &= \int_{A_1^c \cap A_2^c} |q_\gamma(y, x) - q_0(x, y)| \frac{\sqrt{\pi(y)}}{\sqrt{\pi(x)}} dy \leq \int |q_\gamma(y, x) - q_0(x, y)| dy, \end{aligned}$$

since $\frac{\pi(y)}{\pi(x)} \leq 1$ on $A_1^c \cap A_2^c$.

$$\frac{I_2(x, \gamma)}{V(x)} = \int_{A_1 \cap A_2} |q_\gamma(x, y) - q_0(x, y)| \frac{V(y)}{V(x)} dy \leq \int |q_\gamma(x, y) - q_0(x, y)| dy$$

since $\frac{V(y)}{V(x)} < 1$ on A_1 .

$$\begin{aligned} \frac{I_3(x, \gamma)}{V(x)} &= \int_{A_1 \cap A_2^c} \left| \frac{\pi(y)}{\pi(x)} q_\gamma(y, x) - q(x, y) \right| \frac{V(y)}{V(x)} dy \\ &\leq \int_{A_1 \cap A_2^c} |q_\gamma(y, x) - q_0(x, y)| dy + \int_{A_1 \cap A_2^c} q_\gamma(y, x) \left(\frac{\pi(y)}{\pi(x)} - 1 \right) dy \\ &\leq \int |q_\gamma(y, x) - q_0(x, y)| dy + \int |q_\gamma(x, y) - q_\gamma(y, x)| dy, \end{aligned}$$

since on $A_1 \cap A_2^c$, $\frac{V(y)}{V(x)} < 1$, $0 < \frac{\pi(y)}{\pi(x)} - 1 \leq \frac{q_\gamma(x, y) - q_\gamma(y, x)}{q_\gamma(y, x)}$.

Finally,

$$\begin{aligned}
\frac{I_4(x, \gamma)}{V(x)} &= \int_{A_1^c \cap A_2} \left| q_\gamma(x, y) - \frac{\pi(y)}{\pi(x)} q_0(x, y) \right| \frac{V(y)}{V(x)} dy \\
&\leq \int_{A_1^c \cap A_2} |q_\gamma(x, y) - q_0(x, y)| \frac{\sqrt{q_\gamma(y, x)}}{\sqrt{q_\gamma(x, y)}} dy \\
&+ \int_{A_1^c \cap A_2} q_0(x, y) \left(1 - \frac{\pi(y)}{\pi(x)} \right) \frac{\sqrt{q_\gamma(y, x)}}{\sqrt{q_\gamma(x, y)}} dy \\
&\leq \int |q_\gamma(x, y) - q_0(x, y)| \frac{\sqrt{q_\gamma(y, x)}}{\sqrt{q_\gamma(x, y)}} dy \\
&+ \int_{A_1^c \cap A_2} q_0(x, y) \left(1 - \frac{q_\gamma(x, y)}{q_\gamma(y, x)} \right) \frac{\sqrt{q_\gamma(y, x)}}{\sqrt{q_\gamma(x, y)}} dy,
\end{aligned}$$

where we used that on $A_1^c \cap A_2$, $\frac{V(y)}{V(x)} < \frac{\sqrt{q_\gamma(y, x)}}{\sqrt{q_\gamma(x, y)}}$ and $0 \leq 1 - \frac{\pi(y)}{\pi(x)} < \frac{q_\gamma(y, x) - q_\gamma(x, y)}{q_\gamma(y, x)}$.

Because of the bound (3.61), it is easy to verify, using Lebesgue dominated convergence theorem, that for every $\varepsilon > 0$ and compact set Γ , there exists $T < \infty$ such that for $i \in \{1, 2, 3, 4\}$,

$$\sup_{\gamma \in \Gamma, |x| > T} \frac{I_i(x, \gamma)}{V(x)} < \varepsilon.$$

□

3.8 Appendix B

Proof of Lemma 6. The lemma follows from Beardon [1996]. See formula (2.3) therein. Here we provide an alternative proof. We apply Stolz-Cesàro theorem (see Section 3.1.7 of Mureşan [2009]) in order to get

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n i^\beta}{n^{1+\beta}} = \lim_{n \rightarrow \infty} \frac{n^\beta}{n^{1+\beta} - (n-1)^{1+\beta}}.$$

After simple manipulations we get

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{n^\beta}{n^{1+\beta} - (n-1)^{1+\beta}} &= \lim_{n \rightarrow \infty} \frac{1/n}{1 - (1 - 1/n)^{1+\beta}} = \\
&= \lim_{x \rightarrow 0} \frac{x}{1 - (1-x)^{1+\beta}} = \frac{1}{1+\beta},
\end{aligned}$$

where we used L'Hopital's rule to derive the last equality.

□

Proof of Lemma 7. We exploit the proof of Theorem 5 of [Lai and Siegmund \[1979\]](#). Let T_k be the k -th regeneration time of a Markov chain with kernel P_γ started from the regeneration measure ν . Either (3.11), (3.9), (3.10) together with Theorem 4.2 and 5.2 of [Łatuszyński et al. \[2013a\]](#) yield

$$\sigma^2 = \sup_{\gamma \in \Gamma} \mathbb{E}_{\nu, \gamma} T^2 < \infty.$$

To shorten notations, let $\mathbb{E} := \mathbb{E}_{\nu, \gamma}$. The second Wald's identity yields

$$\mathbb{E}[T_{R(b)} - \mu R(b)]^2 = \mathbb{E} T^2 \mathbb{E} R(b).$$

Bounds (3.12) - (3.14) of [Łatuszyński et al. \[2013a\]](#) imply

$$\mathbb{E}[T_{R(b)} - b] \leq 2\mu - 1,$$

$$\mathbb{E} R(b) = \frac{1}{\mu} (b + \mathbb{E}[T_{R(b)} - b]) \leq \frac{1}{\mu} (b + 2\mu - 1).$$

Therefore, we can estimate

$$\begin{aligned} \mathbb{E} |\mu R(b) - b| &= \mathbb{E} |(\mu R(b) - T_{R(b)}) + (T_{R(b)} - b)| \\ &\leq \sqrt{\mathbb{E} [\mu R(b) - T_{R(b)}]^2} + \mathbb{E} [T_{R(b)} - b] \\ &\leq \sqrt{\mathbb{E} T^2 \mathbb{E} R(b)} + 2\mu - 1 \leq \sigma \sqrt{\frac{1}{\mu} (b + 2\mu - 1)} + 2\mu - 1, \end{aligned}$$

which finishes the proof.

□

Proof of Lemma 8. Follows immediately from the proof of Theorem 3 of [Roberts and Rosenthal \[2007\]](#).

□

Proof of Lemma 9. The inequality (3.43) is derived in Theorem 10 of [Bai et al. \[2011\]](#), where it is shown, in particular, that there exists constant M_1 such that for all $n, \xi \in [1, 1/(1 - \alpha))$, and large m ,

$$\mathbb{P}_{(X_0, \gamma_0)} \left(V^{1-\xi(1-\alpha)}(X_n) > m \right) \leq M_1(1 + V(X_0)) \sum_{i=0}^{n-1} \frac{1}{(n-i)^{\xi-1}(m+n-i)}.$$

Since $\alpha \geq 2/3$ by the conditions of Theorem 12, we can take $\xi = 2$ and obtain the following bound

$$\mathbb{P}_{(X_0, \gamma_0)} \left(V^{2\alpha-1}(X_n) > m \right) \leq M_1(1 + V(X_0)) \sum_{i=0}^{n-1} \frac{1}{(n-i)(m+n-i)}.$$

Integral convergence test for series (see Chapter 23 of Spivak [1994]) implies that for all $n > 1$, $\sum_{i=0}^{n-1} \frac{1}{(n-i)(m+n-i)}$ is bounded by $\frac{\log(1+m)}{m} + \frac{1}{m+1}$ which proves (3.43). \square

Proof of Lemma 10. Using Jensen's inequality, we get

$$\mathbb{E}_{\nu, \gamma} \left[\sum_{j=0}^{T-1} f(X_j) \right]^{2+\delta} \leq \mathbb{E}_{\nu, \gamma} \left[T^{p-1} \sum_{j=0}^{T-1} f(X_j)^p \right]^{\frac{2+\delta}{p}},$$

Now Hölder inequality yields

$$\begin{aligned} \mathbb{E}_{\nu, \gamma} \left[T^{p-1} \sum_{j=0}^{T-1} f(X_j)^p \right]^{\frac{2+\delta}{p}} &= \mathbb{E}_{\nu, \gamma} \left[T^{\frac{(p-1)(2+\delta)}{p}} \left(\sum_{j=0}^{T-1} f(X_j)^p \right)^{\frac{2+\delta}{p}} \right] \\ &\leq \left(\mathbb{E}_{\nu, \gamma} \left[T^{\frac{(2+\delta)(p-1)}{p-2-\delta}} \right] \right)^{\frac{p-2-\delta}{p}} \left(\mathbb{E}_{\nu, \gamma} \left[\sum_{j=0}^{T-1} f(X_j)^p \right] \right)^{\frac{2+\delta}{p}}. \end{aligned}$$

\square

Chapter 4

Software package

4.1 Overview

Metropolis-Hastings and Gibbs Sampler Algorithms [1](#) and [2](#) are arguably two of the most widely used Markov Chain Monte Carlo (MCMC) methods. The popularity of the algorithms is explained by the simplicity of their implementation and analysis.

In this chapter we describe a software package [Chimisov \[2018\]](#), where we implement the ARSGS and ARWMwAG Algorithms [9](#) and [12](#) presented in Chapter [2](#). The software is open source and freely available on GitHub. The package is closely related to the AMCMC package of [Rosenthal \[2007\]](#) that implements the Adaptive Metropolis within Gibbs algorithm in C providing R interface. We extend the AMCMC package in a multiple ways. First, we rewrite the code in C++ using Rcpp library to make the source code more user-friendly. Secondly, the adaptive algorithms tune both the variance of the proposal and the selection probabilities of the RWM algorithm. Thirdly, we allow the user to employ blocking schemes for the algorithm (see, e.g., [Roberts and Sahu \[1997a\]](#)). Fourthly, the package supports the ARSGS Algorithm [9](#) for user-defined sampling procedures from the full conditional distributions. Finally, we implement Air versions of the algorithms, i.e., allow for an increasing lag between adaptations (see Section [3.4.1](#) of Chapter [3](#)).

At a high-level, the user can run the ARWMwAG Algorithm [12](#) by solely specifying the target density in either R or C++ language. At a low level, the user may specify sampling procedure from full-conditionals in C++ and perform the ARSGS. The user can also use a mixture of the AMwAG and ARSGS, where some of the components are updated using either Metropolis normal proposals or full-conditional updates.

Detailed manual is provided on the package source code web-page [Chimisov](#)

[2018]. We shall only outline the main features of the package.

In Section 4.2 we briefly guide through the installation instructions to the library. We shall explain how to specify the target density in R and C++ in Sections 4.3 and 4.4, respectively. We shall demonstrate usage of the library in Section 4.5 by sampling from the normal distribution with a toy covariance matrix presented in Example 1 of Section 2.3. Various library features are described in Sections 4.6, 4.7, and 4.8.

Running example. Throughout, we will demonstrate performance on sampling from a d -dimensional multivariate normal distribution with mean zero and covariance matrix has blocking structure like in Example 1 of Section 2.3, Chapter 2, namely, $\Sigma = (\Sigma_{ij})_{i,j=1}^d$, where $\Sigma_{ii} = 1$, $\Sigma_{2i-1,2i} = \Sigma_{2i,2i-1} = \frac{-0.95}{i}$, and $\Sigma_{ij} = 0$ otherwise.

4.2 Installation

The library consists of a series of R-callables. The dependencies are Rcpp, RcppArmadillo, and RcppParallel libraries.ⁱ If not yet installed, the user is required to do so through an R session. Rcpp provides integration between C++ and R, RcppArmadillo is a linear algebra library, and RcppParallel is used for the parallel adaptation feature described later in Section 4.6.6.

4.2.1 Compile

In order to use the library functions, the user is required to manually compile the library. For this purpose, copy the library to a desired folder and in the library folder run `sourceCpp` for the “Adaptive_Gibbs.cpp” file:

```
Rcpp::sourceCpp("Adaptive_Gibbs.cpp")
```

4.2.2 After compilation

Create a folder where the output of the chain will be stored, say, “../simulation_results”. Set this folder to be the working directory using `set_working_directory` command which is defined in Adaptive_Gibbs.hpp:

```
set_working_directory("../simulation_results/")
```

The main function provided with the library is `AMCMC(...)` which performs sampling and saves the output to the folder defined by `set_working_directory`

command. The full list of arguments is available in a package description files provided in [Chimisov \[2018\]](#).

4.3 R-defined target densities

We recommend to define the density in C++ since the program will execute much faster in that case. However, for simple problems, it should suffice to define the target distribution in R. Notice, only the MwG with normal proposals and its adaptive versions are implemented for R-defined densities.

As a simple example, consider sampling from normal target distribution with precision matrix Q . For demonstration purposes “gaussian_target.hpp” provides `set_example_covariance` function that takes the number of dimensions as an argument and produces the block diagonal matrix Σ as described in the running example of Section 4.1.

```
dim <- 10 # dimensionality of the target
N <- 10000 # number of desired samples
# set a covariance matrix such as in the Introduction
set_example_covariance(dim)
# set precision matrix. Here get_covariance returns the value of
↪ set_example_covariance(dim)
Q = solve(get_covariance())
# set logarithm of the target distribution up to a normalising
↪ constant. Should return a real number
example_logdensity<-function(x)
{
  return( -1./2* (t(x) %*%Q%*% x)[1,1] )
}
# set random seed
set.seed(1)
#run adaptive MCMC for N steps
adaptive_chain <- AMCMC(R_density = example_logdensity, logdensity =
↪ 1, dimension = dim, N = N)
```

Here `AMCMC(..)` performs the coordinate-wise ARWMwAG Algorithm 12 with normal proposals described in Section 2.5 of Chapter 2. `dim` and `N` denote the number of dimensions and the number of desired samples, respectively. Non-empty parameter `R_density` indicates that R-defined density is used. Setting

`logdensity = 1` means that the input `R_density` is passed as a logarithm of the target density function.

Function `AMCMC(...)` returns estimated values for the pseudo-spectral gap and optimal sampling probabilities (as defined in Section 2.2 of Chapter 2):

```
> adaptive_chain
$sp_gap
[1] 0.02061687

$weights
[1] 0.37989888 0.37542078 0.04652505 0.04120361 0.02780708
↪ 0.03084711 0.01387269 0.03186175
[9] 0.02678018 0.02578287
```

The user can trace output of each coordinate using `trace_coordinate(i)` command. For example, in order to obtain the estimated correlation matrix, we could run the following script:

```
S <- matrix(0,nrow = N, ncol = dim)
for(i in 1:dim)
{
  v <- trace_coord(i)
  S[,i] <- v
}
S <- cor(S)
# estimated correlation matrix:
S
```

4.4 C++-defined target densities

As we have stated in the previous section, the current library is designed to run adaptive MCMC algorithms for C++-defined densities. The main drawback as such is that the user has to manually recompile the whole library every time a new C++-density is defined. For convenience, we provide a “template.hpp” file, where the target density could be specified. The target density is treated as a class. In “template.hpp” file we can see that the density class has the following structure:

```

class new_density: public distribution_class
{
public:
    //add some additional functions/parameters if necessary
    new_density();//specify constructor if necessary
    ~new_density(); //specify destructor if necessary
    double density(vec theta);    //target density at the point
    ↪ theta
    double logdensity(vec theta); //logarithm of the target
    ↪ density
    double full_cond(vec theta, int block_ind); //full
    ↪ conditional of the block `block_ind`
    double logfull_cond(vec theta, int block_ind); //logarithm
    ↪ of the full conditional of the block `block_ind`
    vec sample_full_cond(vec theta, int block_ind); //sample
    ↪ block `block_ind` from its full conditional
    ↪ distributiont
};

```

That is, a new user-defined density should be a child of a general parent abstract class `distribution_class` defined in “Adaptive-Gibbs.hpp”. In order to sample from the new distribution, the user is expected to specify at least one of the attributes.

As an example, we specified all the attributes for normal target distribution in “gaussian_target.hpp file”. Note that specifying the log-density is extremely easy in this case:

```

double gaussian::logdensity(vec theta)
{
    return -1./2*dot(theta.t()*Q,theta);
}

```

Here Q is a precision matrix that is defined in the class constructor, and `dot(..)` is a dot product provided with RcppArmadillo library.

After the density (or log-density) functions are specified in “template.hpp”, the minimum code to start sampling is

```

AMCMC(distribution_type = "new_density", dimension = dim, N = N)

```

Argument `distribution_type = "new_density"` tells the `AMCMC(..)` function to use the distribution defined in “template.hpp”.

As an example, the following script can be used for sampling from the multi-variate normal distribution with the toy covariance matrix described in the running example of Section 4.1:

```
# set a covariance matrix such as in the Introduction
set_example_covariance(dim)
AMCMC(distribution_type = "gaussian", logdensity = 1, dimension =
  ↪ dim, N = N)
```

In Section 4.8 we describe how to customize “template.hpp” file in order to create a new name for `distribution_type` parameter.

4.5 Does the adaptation help?

In this section we demonstrate potential speed up of the adaptive algorithms. We shall run three different algorithms for the running example described in Section 4.1 and compare the output.

Consider a 10-dimensional target normal distribution centred at the origin.

```
dim <- 10 # dimensionality of the target
N <- 10000 # number of desired samples
# set a covariance matrix such as in the Introduction
set_example_covariance(dim)
# set random seed
set.seed(42)
```

We run the RWMwG started at (100,..,100) with misspecified starting directional proposal variances (0.1,..,0.1).

```
AMCMC(distribution_type = "gaussian", logdensity = 1, dimension =
  ↪ dim, N = N, adapt_weights = 0, adapt_proposals = 0,
  ↪ start_location = rep(100, 10), start_scaling = rep(0.1,10))
```

We have disabled adaptations completely by setting `adapt_weights = 0`, `adapt_proposals = 0` (see Section 4.6.2 for details). We can access the output of every coordinate `i` by invoking `trace_coord(i)`. Since in this case the covariance of the target distribution is known and has a block-diagonal form, we trace only

1st and 10th coordinate (the “hardest” and the “easiest ” coordinates). On Figure 4.1 we produce trace plots for the first 15000 iterations after the burn-in and the estimated autocorrelation (ACF) plot. It can be seen that the MwG with badly tuned parameters converges extremely slowly.

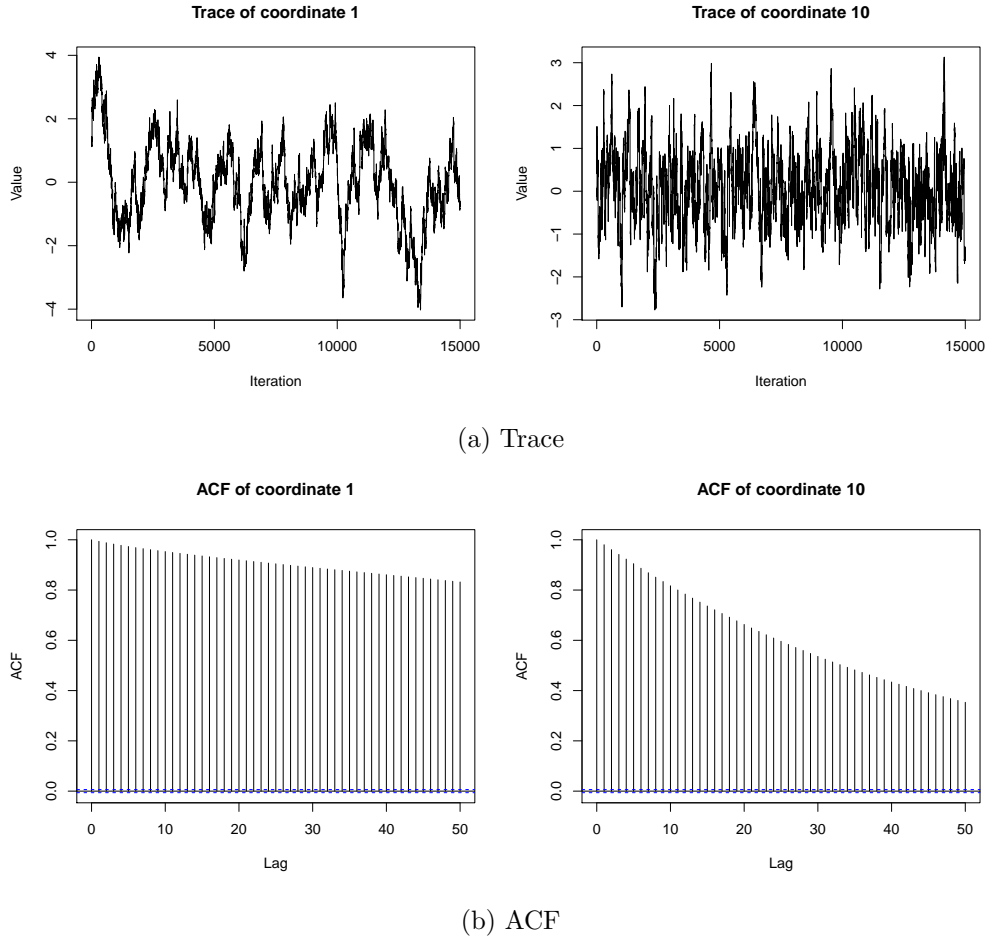
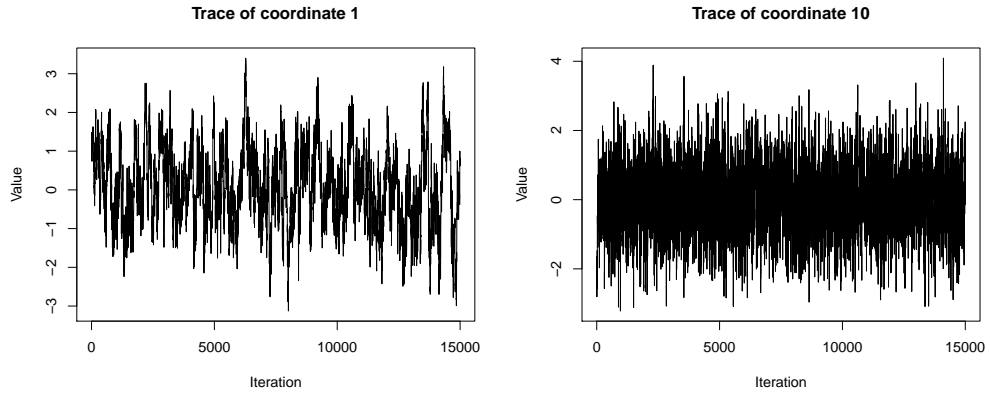
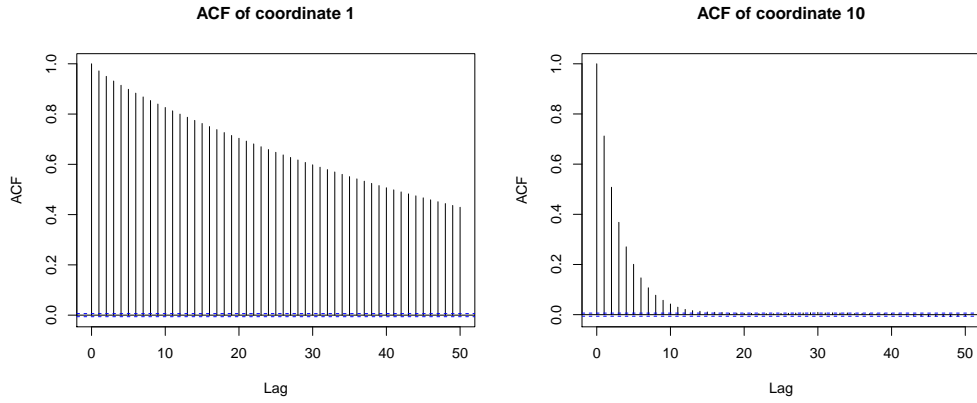


Figure 4.1: Metropolis within Gibbs

On the other hand, by adapting the proposals (set `adapt_proposals = 1`) as in the ARWMwG Algorithm 11, convergence can be significantly improved as seen from Figure 4.2



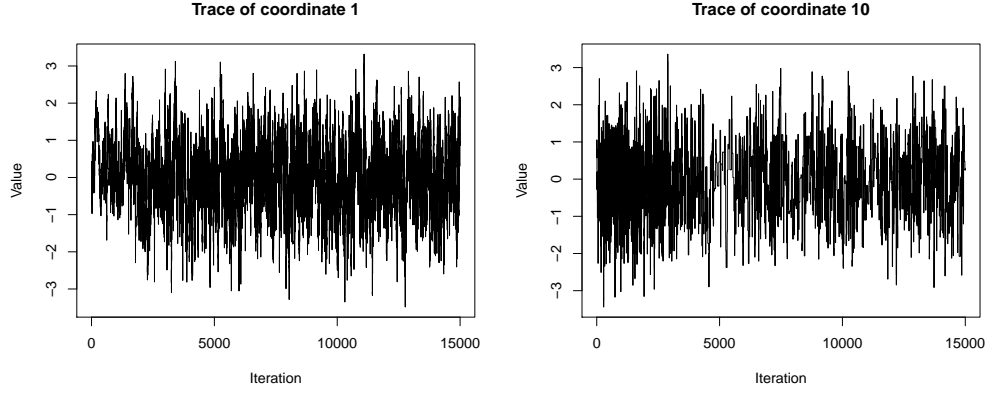
(a) Trace



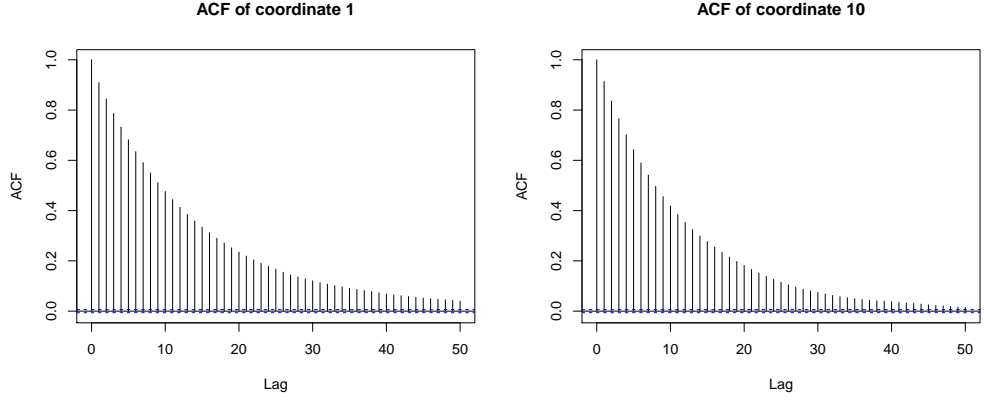
(b) ACF

Figure 4.2: Adaptive Random Walk Metropolis within Gibbs

The autocorrelation plot 4.2b suggests that convergence along the first coordinate can be improved by putting more selection probability weight on it. By setting `adapt_weights = 1` and `adapt_proposals = 1` we run the fully adaptive ARWMwAG Algorithm 12. The corresponding plots for 1st and 10th coordinates can be seen on Figure 4.3.



(a) Trace



(b) ACF

Figure 4.3: Adaptive Random Walk Metropolis within Adaptive Gibbs

Roughly speaking, for a target distribution π , the ARWMwAG sampler minimises convergence time of the chain output average (1.1) to $\int f(x)d\pi(x)$ for the worst case function f .

One can trace the sampling weights by invoking `trace_weights()`. It is easy to conclude then that the optimal sampling weights are found only after 40000 iterations of the adaptive chain. However, as one can see from the trace plots on Figure 4.3a, even when the optimal parameters are not found yet, the chain successfully manages to explore the target probability distribution even within first 15000 iterations.

4.6 AMCMC function

In this section we shall describe some of the main arguments of `AMCMC(...)`. The full description is available on the package documentation web page [Chimisov \[2018\]](#).

4.6.1 Starting values

If not specified, the starting location `start_location` is set to be the origin (i.e., a vector of zeros), starting sampling weights `start_weights` are uniform $\frac{1}{\text{dim}}$, and starting scalings `start_scaling` are set to be ones. If parameter `blocking` is not specified, then `start_scaling` is exactly the starting value for the standard deviation of the normal proposals.

Burn-in parameter `frac_burn_in = 10` is by default set to be 10 percent of the length of the desired sample size N .

4.6.2 Adaptations

By default, both sampling weights and Metropolis proposal variances are adapted. The adaptation procedure is implemented in a way described in Algorithms 9 and 12 of presented in Chapter 2. If the user doesn't want to adapt the proposal scalings or the sampling weights, the corresponding parameters `adapt_weights` and `adapt_proposals` may be disabled. In order to disable estimation of the pseudo-spectral gap introduced in Section 2.2 of Chapter 2, one needs to set `estimate_spectral_gap = 0`. Also, the number of iterations between adaptations may be tuned. Too frequent adaptation may impose significant additional computational burden (see Section 4.6.6 for details). Parameter `batch_length` indicates number of iterations of a Markov chain before changing the sampling weights, while `frequency_proposal_update` controls the relative frequency of scales adaptations. By default, `batch_length=100` and `frequency_proposal_update = 100`, which means that proposals are updated at each iteration of the algorithm.

4.6.3 Blocking

By default, the coordinate-wise AMwAG is performed. Argument `blocking` allows the user to specify a blocking structure for the algorithm. For example, if `dimension = 10` and `blocking = c(5, 2, 3)`, the coordinates are grouped into three blocks: first 5 coordinates are combined into a first block, coordinates 6 and 7 into a second block, and the last three coordinates into a third block. The algorithm then updates all the coordinates within a block simultaneously.

At iteration n , a normal proposal $N(X_n, \sigma_{n,i}^2 \Sigma_{n,i})$ is generated for a block i and then accepted/rejected using Metropolis procedure. If the block is of size one, $\Sigma_{n,i} := 1$ and the corresponding value of the scaling parameter $\sigma_{n,i}^2$ is tuned in order to retain average acceptance ratio around 0.44. If the block size is greater than one, the proposal scaling $\sigma_{n,i}^2$ is tuned to achieve the average acceptance ration of 0.234, whereas the covariance $\Sigma_{n,i}$ is set to be Q_{ii}^{-1} . Here Q is the estimated precision matrix (i.e., inverse of the covariance) and Q_{ii} is the i -th diagonal block. Note, if the target distribution is normal, Q_{ii}^{-1} represent conditional covariance matrix of the block i and our choice of proposal is motivated by [Haario et al. \[2001\]](#), [Rosenthal \[2011\]](#).

We also provide a methodological improvement for the sampling weights tuning procedure. Say, there are s blocks to update of size d_i each. Heuristically, the standard Metropolis-Hastings algorithm with full dimensional normal proposal requires $\mathcal{O}(\text{dim})$ iterations to converge (see e.g., [Green et al. \[2015\]](#)). That is, larger blocks i require d_i times more iterations to achieve the same convergence rate as the 1-dimensional blocks. In order to account for this fact, having estimated the optimal sampling weights to be, say, (p_1, \dots, p_s) , we re-weight the selection probabilities by multiplying them by the size of the corresponding blocks. Namely, we offer to sample the next block using the adjusted selection

$$\frac{1}{C}(p_1 d_1, \dots, p_s d_s),$$

where $C = \sum_{i=1}^s p_i d_i$ is a normalising constant.

The user can enable the re-weighting procedure by setting `reweight = 1`.

4.6.4 Full conditional density specification

This feature is available for C++-defined densities only. Argument `full_cond`, if enabled, tells the algorithm to use `double full_cond(vec theta, int block_ind)` function from “Adaptive_Gibbs.hpp” that is used to compute the Metropolis-Hastings acceptance ratio. This function allows the user to specify the full conditional density directly. Here `theta` is a point in a state space and `block_ind` is a block index to be updated.

The blocking structure of the algorithm is recorded in a global variable `blocking_structure`. In order to access block `block_ind` of the vector `theta`, one can refer to its corresponding subvector using `blocking_structure`. For example,

```
block(block_ind) = theta.subvec(blocking_structure(block_ind),
↪ blocking_structure(block_ind+1)-1)
```

provides access to the block `block_ind` of vector `theta`. A comprehensive example is presented in “gaussian_target.hpp” file, where the full conditional density is described as:

```
double gaussian::full_cond(vec theta, int block_ind)
{
    vec mn(blocking_structure(block_ind + 1) -
            blocking_structure(block_ind));
    vec v(blocking_structure(block_ind + 1) -
            blocking_structure(block_ind));
    mn.zeros();
    for (int j = 0; j<par; j++)
    {
        mn = (mn + A_block(block_ind, j) *
                theta.subvec(blocking_structure(j),
                            blocking_structure(j+1)-1));
    }
    v = (inv_sd(block_ind) *
            theta.subvec(blocking_structure(block_ind),
                        blocking_structure(block_ind+1)-1)
        - mn);
    return exp(-1./2*dot(v,v));
}
```

Here `par` is a global variable of the number of blocks in the Gibbs sampling scheme. We tried to be consistent with notations of Roberts and Sahu [1997a]. Variable `mn` represents the conditional mean. Matrix valued functions `inv_sd` and `A_block` are described in the constructor of the class. `inv_sd(block_ind)` represents conditional precision matrix of the block `block_ind`, whereas `A_block` is the matrix A in formula (3) of Roberts and Sahu [1997a] defined as $A_block = I - \text{diag}(Q_{11}^{-1}, \dots, Q_{ss}^{-1})Q$.

If the parameter `logdensity` is enabled, the full conditional log-density `double logfull_cond(vec theta, int block_ind)` is utilised by the algorithm.

4.6.5 Gibbs sampling

For C++-defined densities we have also provided Gibbs sampling support if the user can specify the sampling procedure from the full conditional distribution. In order to enable Gibbs sampling updates, one has to set `gibbs_sampling = 1`. Moreover, one can provide `gibb_step` vector of 0s and 1s, where 1 indicates that the corresponding block is updated using Gibbs sampling. For example, for a given blocking structure, say `blocking = c(1,2,2,5)`, `gibb_step = c(1,1,0,0)` indicates that the first two blocks are updated using Gibbs sampling and the last two are updated using the Metropolis-Hastings procedure with normal proposals.

In order to describe Gibbs sampling step for a block `block_ind`, one has to specify `vec sample_full_cond(vec theta, int block_ind)` function in “Adaptive-Gibbs.hpp”. The function should return a vector of the same length as the length of block `block_ind`. Notice that the user is expected to use R random numbers generator. More precisely, any R random generating function that is supported Rcpp could be used, e.g., `rnorm`, `rt`, `runif`, `rbeta`, etc. A comprehensive example from `gaussian.target.hpp` file is below.

```
vec gaussian::sample_full_cond(vec theta, int block_ind)
{
    vec res(blocking_structure(block_ind + 1) -
            blocking_structure(block_ind));
    res.zeros();
    for (int j = 0; j<par; j++)
    {
        res = (res + A_block(block_ind, j) *
                theta.subvec(blocking_structure(j),
                            blocking_structure(j+1)-1));
    }
    res = (res + sd(block_ind) *
            vec(rnorm(blocking_structure(block_ind+1) -
                    blocking_structure(block_ind)))));
    return res;
}
```

Note that `rnorm(n)` generates an Rcpp vector of length n of type `NumericVector`, which is then converted to an Armadillo vector using `vec(...)`.

4.6.6 Parallel adaptations

It is possible to perform sampling and adaptations simultaneously as we have discussed in Section 2.7.3 of Chapter 2. By doing so, we could suppress the additional computational time imposed by adaptations. To enable the parallel computation feature, one has to set `parallel_adaptation = 1`. It will often be the case that the total adaptation time is negligible compared to the total computational effort. In other cases, as we have seen in Section 3.1 of Chapter 3, adaptation time may not be ignored.

Below we present computational script that compares time spent for sampling from a 500 dimensional normal distribution from the running example of Section 4.1.

```
dim <- 500
N <- 10000
set_example_covariance(dim)

# Time spent without adaptations
system.time(AMCMC(distribution_type = "gaussian", dimension = dim, N
↪ = N, full_cond = 1, blocking = c(rep(1,100), rep(5,80)), save =
↪ 0, adapt_proposals = 0, adapt_weights = 0, estimate_spectral_gap
↪ = 0))
  user  system elapsed
54.959   0.165   55.130

# Time spent without parallelisation
system.time(AMCMC(distribution_type = "gaussian", dimension = dim, N
↪ = N, full_cond = 1, blocking = c(rep(1,100), rep(5,80)), save =
↪ 0, parallel_adaptation = 0))
  user  system elapsed
93.459   6.929  100.469

# Time spent with parallelisation
system.time(AMCMC(distribution_type = "gaussian", dimension = dim, N
↪ = N, full_cond = 1, blocking = c(rep(1,100), rep(5,80)), save =
↪ 0, parallel_adaptation = 1))
  user  system elapsed
103.766   6.883   68.123
```


4.6.7 AirMCMC

The current implementation supports Air versions of the algorithms (see Algorithm 4 in Section 1.3.2 of Chapter 1). In order to enable the AirMCMC, the user has to set `rate_beta` to any positive real number. This will iteratively change `batch_length` to `batch_length * floor (i^{rate_beta})` after sampling i -th batch of samples. Note that the initial value of `batch_length` can be any positive real number which is rounded to the nearest positive integer during the sampling stage of `AMCMC(...)` function.

4.7 Accessing chain output

We provide a series of commands to access the chain output. In order to trace a coordinate i one can use `trace_coord(i)` command, as we have seen in an example from Sections 4.3, where this function was used to estimate the correlation matrix. Moreover, the user is free to trace adapted probability weights, spectral gap, and scalings of the proposals.

```
Prob <- trace_weights()
# trace of the estimated optimal weights:
Prob
#####
Gap <- trace_inv_sp_gap()
# trace of the estimated 1/spectral gap:
Gap
#####
Scaling <- trace_proposals()
# trace of the estimated optimal proposal scalings:
Scaling
```

4.8 Customising template.hpp

One can manually extend the scope of possible distribution type names. Say, the user wants to describe a density “my_distribution” and being able to call directly

```
AMCMC(distribution_type = "my_distribution", ... )
```

This can be done through the following steps:

1. Create a new .hpp file and call it, say, “my_distribution.hpp”.
2. Copy paste the content of “template.hpp” to the newly created file “my_distribution.hpp”.
3. Change the class name from “new_density” to “my_distribution”.
4. Define the desired functions of the class “my_distribution”.
5. Open “density_list.hpp” file. In the preamble add

```
#include "my_distribution.hpp"
```

and also add the following to `density_list` function:

```
else if(distribution_type == "t_distribution")
{
    c_density = new my_distribution;
}
```

4.9 Discussion

The Adaptive Gibbs library of [Chimisov \[2018\]](#) provides C++ implementation of the ARSGS and ARWMwAG algorithms presented in Chapter 2 and their Air versions as described in Chapter 3. We provide an R interface allowing C++-inexperienced users running standard and adaptive versions of popular MCMC algorithms. A number of examples are included in accompanying tutorial files, which should help the users avoid any confusions when working with the library.

Bibliography

- Yali Amit. On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions. *J. Multivariate Anal.*, 38(1):82–99, 1991. ISSN 0047-259X. doi: 10.1016/0047-259X(91)90033-X. URL [http://dx.doi.org/10.1016/0047-259X\(91\)90033-X](http://dx.doi.org/10.1016/0047-259X(91)90033-X).
- Yali Amit. Convergence properties of the Gibbs sampler for perturbations of Gaussians. *Ann. Statist.*, 24(1):122–140, 1996. ISSN 0090-5364. doi: 10.1214/aos/1033066202. URL <http://dx.doi.org/10.1214/aos/1033066202>.
- Alan L. Andrew, K.-w. Eric Chu, and Peter Lancaster. Derivatives of eigenvalues and eigenvectors of matrix functions. *SIAM J. Matrix Anal. Appl.*, 14(4):903–926, 1993. ISSN 0895-4798. URL <https://doi.org/10.1137/0614061>.
- George E. Andrews, Richard Askey, and Ranjan Roy. *Special functions*, volume 71 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1999. ISBN 0-521-62321-9; 0-521-78988-5. URL <https://doi.org/10.1017/CB09781107325937>.
- Christophe Andrieu and Yves F. Atchadé. On the efficiency of adaptive MCMC algorithms. *Electron. Comm. Probab.*, 12:336–349 (electronic), 2007. ISSN 1083-589X. doi: 10.1214/ECP.v12-1320. URL <http://dx.doi.org/10.1214/ECP.v12-1320>.
- Christophe Andrieu and Éric Moulines. On the ergodicity properties of some adaptive MCMC algorithms. *Ann. Appl. Probab.*, 16(3):1462–1505, 2006. ISSN 1050-5164. URL <https://doi.org/10.1214/105051606000000286>.
- Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Stat. Comput.*, 18(4):343–373, 2008. ISSN 0960-3174. URL <https://doi.org/10.1007/s11222-008-9110-y>.

- Yves Atchadé and Gersende Fort. Limit theorems for some adaptive MCMC algorithms with subgeometric kernels. *Bernoulli*, 16(1):116–154, 2010. ISSN 1350-7265. doi: 10.3150/09-BEJ199. URL <http://dx.doi.org/10.3150/09-BEJ199>.
- Yves Atchadé, Gersende Fort, Eric Moulines, and Pierre Priouret. Adaptive Markov chain Monte Carlo: theory and methods. In *Bayesian time series models*, pages 32–51. Cambridge Univ. Press, Cambridge, 2011.
- Yves F. Atchadé. An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodol. Comput. Appl. Probab.*, 8(2):235–254, 2006. ISSN 1387-5841. URL <https://doi.org/10.1007/s11009-006-8550-0>.
- Yves F. Atchadé and Jeffrey S. Rosenthal. On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11(5):815–828, 2005. ISSN 1350-7265. doi: 10.3150/bj/1130077595. URL <http://dx.doi.org/10.3150/bj/1130077595>.
- Yan Bai, Gareth O. Roberts, and Jeffrey S. Rosenthal. On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Adv. Appl. Stat.*, 21(1):1–54, 2011. ISSN 0972-3617.
- AA Barker. Monte carlo calculations of the radial distribution functions for a proton?electron plasma. *Australian Journal of Physics*, 18(2):119–134, 04 1965. URL <https://doi.org/10.1071/PH650119>.
- A. F. Beardon. Sums of powers of integers. *Amer. Math. Monthly*, 103(3):201–213, 1996. ISSN 0002-9890. doi: 10.2307/2975368. URL <http://dx.doi.org/10.2307/2975368>.
- Mylène Bédard. Weak convergence of Metropolis algorithms for non-i.i.d. target distributions. *Ann. Appl. Probab.*, 17(4):1222–1244, 2007. ISSN 1050-5164. URL <https://doi.org/10.1214/105051607000000096>.
- Mylène Bédard and Jeffrey S. Rosenthal. Optimal scaling of Metropolis algorithms: heading toward general target distributions. *Canad. J. Statist.*, 36(4):483–503, 2008. ISSN 0319-5724. URL <https://doi.org/10.1002/cjs.5550360401>.
- Witold Bednorz and Krzysztof Łatuszyński. A few remarks on “Fixed-width output analysis for Markov chain Monte Carlo” by Jones et al. [mr2279478]. *J. Amer. Statist. Assoc.*, 102(480):1485–1486, 2007. ISSN 0162-1459. URL <https://doi.org/10.1198/0162145070000000914>.

- Witold Bednorz, Krzysztof Łatuszyński, and Rafał Łatała. A regeneration proof of the central limit theorem for uniformly ergodic Markov chains. *Electron. Commun. Probab.*, 13:85–98, 2008. ISSN 1083-589X. URL <https://doi.org/10.1214/ECP.v13-1354>.
- Dimitri P. Bertsekas. *Convex analysis and optimization*. Athena Scientific, Belmont, MA, 2003. ISBN 1-886529-45-0. With Angelia Nedić and Asuman E. Ozdaglar.
- Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 2013. ISSN 1350-7265. URL <https://doi.org/10.3150/12-BEJ414>.
- Peter J. Bickel and Elizaveta Levina. Covariance regularization by thresholding. *Ann. Statist.*, 36(6):2577–2604, 2008. ISSN 0090-5364. URL <https://doi.org/10.1214/08-AOS600>.
- Joris Bierkens and Andrew Duncan. Limit theorems for the zig-zag process. *Adv. in Appl. Probab.*, 49(3):791–825, 2017. ISSN 0001-8678. URL <https://doi.org/10.1017/apr.2017.22>.
- Jonathan M. Borwein and Jon D. Vanderwerff. *Convex functions: constructions, characterizations and counterexamples*, volume 109 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2010. ISBN 978-0-521-85005-6. URL <https://doi.org/10.1017/CB09781139087322>.
- Leonard Bottolo and Sylvia Richardson. Evolutionary stochastic search for Bayesian model exploration. *Bayesian Anal.*, 5(3):583–618, 2010. ISSN 1936-0975. URL <https://doi.org/10.1214/10-BA523>.
- Laird Arnault Breyer, Mauro Piccioni, and Sergio Scarlatti. Optimal scaling of MaLa for nonlinear regression. *Ann. Appl. Probab.*, 14(3):1479–1505, 2004. ISSN 1050-5164. URL <https://doi.org/10.1214/105051604000000369>.
- Cyril Chimisov. Adaptive gibbs sampler. <https://github.com/cyrilchim/Adaptive-Gibbs-Sampler/>, 2018.
- King-wah Eric Chu. On multiple eigenvalues of matrices depending on several parameters. *SIAM J. Numer. Anal.*, 27(5):1368–1385, 1990. ISSN 0036-1429. URL <https://doi.org/10.1137/0727079>.

- Radu V. Craiu, Lawrence Gray, Krzysztof Łatuszyński, Neal Madras, Gareth O. Roberts, and Jeffrey S. Rosenthal. Stability of adversarial Markov chains, with an application to adaptive MCMC algorithms. *Ann. Appl. Probab.*, 25(6):3592–3623, 2015. ISSN 1050-5164. doi: 10.1214/14-AAP1083. URL <http://dx.doi.org/10.1214/14-AAP1083>.
- DLMF. *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>, Release 1.0.13 of 2016-09-16. URL <http://dlmf.nist.gov/>. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.
- Hani Doss and B Narasimhan. Bayesian poisson regression using the gibbs sampler: Sensitivity analysis through dynamic graphics. 08 1994.
- Randal Douc, Arnaud Guillin, and Eric Moulines. Bounds on regeneration times and limit theorems for subgeometric Markov chains. *Ann. Inst. Henri Poincaré Probab. Stat.*, 44(2):239–257, 2008. ISSN 0246-0203. URL <https://doi.org/10.1214/07-AIHP109>.
- Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987. ISSN 0370-2693. doi: [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X). URL <http://www.sciencedirect.com/science/article/pii/037026938791197X>.
- Alain Durmus, Eric Moulines, and Eero Saksman. On the convergence of Hamiltonian Monte Carlo. pages 1–32, 2017. URL <http://arxiv.org/abs/1705.00166>.
- Aryeh Dvoretzky. Asymptotic normality for sums of dependent random variables. *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability (Univ. California, Berkeley, Calif., 1970/1971), Vol. II: Probability theory*, pages 513–535, 1972.
- G. Fort and E. Moulines. Polynomial ergodicity of Markov transition kernels. *Stochastic Process. Appl.*, 103(1):57–99, 2003. ISSN 0304-4149. URL [https://doi.org/10.1016/S0304-4149\(02\)00182-5](https://doi.org/10.1016/S0304-4149(02)00182-5).
- Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.*, 85(410):398–409, 1990. ISSN 0162-1459. URL [http://links.jstor.org/sici?sici=0162-1459\(199006\)85:410<398:SATCMD>2.0.CO;2-3&origin=MSN](http://links.jstor.org/sici?sici=0162-1459(199006)85:410<398:SATCMD>2.0.CO;2-3&origin=MSN).

- A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. In *Bayesian statistics, 5 (Alicante, 1994)*, Oxford Sci. Publ., pages 599–607. Oxford Univ. Press, New York, 1996.
- Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian data analysis*. Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2004. ISBN 1-58488-388-X.
- Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596.
- John Geweke. Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints and the Evaluation of Constraint Probabilities. *1991 Computing Science and Statistics: the Twenty-Third Symposium on the Interface*, pages 1–14, 1991. doi: 10.1.1.26.6892.
- Walter R. Gilks, Gareth O. Roberts, and Sujit K. Sahu. Adaptive Markov chain Monte Carlo through regeneration. *J. Amer. Statist. Assoc.*, 93(443):1045–1054, 1998. ISSN 0162-1459. doi: 10.2307/2669848. URL <http://dx.doi.org/10.2307/2669848>.
- W.R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, 1992. ISSN 0035-9254. doi: 10.2307/2347565. URL <http://www.jstor.org/stable/10.2307/2347565>.
- Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 73(2):123–214, 2011. ISSN 1369-7412. URL <https://doi.org/10.1111/j.1467-9868.2010.00765.x>. With discussion and a reply by the authors.
- Peter W. Glynn and Sean P. Meyn. A Liapounov bound for solutions of the Poisson equation. *Ann. Probab.*, 24(2):916–931, 1996. ISSN 0091-1798. URL <https://doi.org/10.1214/aop/1039639370>.
- Peter J. Green, Krzysztof Łatuszyński, Marcelo Pereyra, and Christian P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Stat. Comput.*, 25(4):835–862, 2015. ISSN 0960-3174. URL <https://doi.org/10.1007/s11222-015-9574-5>.

- J. Griffin, K. Łatuszyński, and M. Steel. In Search of Lost (Mixing) Time: Adaptive Markov chain Monte Carlo schemes for Bayesian variable selection with very large p . *ArXiv e-prints*, August 2017.
- Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001. ISSN 1350-7265. doi: 10.2307/3318737. URL <http://dx.doi.org/10.2307/3318737>.
- W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444. URL <https://doi.org/10.1093/biomet/57.1.97>.
- Edwin Hewitt and Karl Stromberg. *Real and Abstract Analysis*. Springer, New York, 1965. ISBN 978-3-662-28275-5. doi: 10.2307/2315158. URL <http://www.jstor.org/stable/2315158?origin=crossref>.
- Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15: 1593–1623, 2014. ISSN 1532-4435.
- Søren F. Jarner and Gareth O. Roberts. Polynomial convergence rates of Markov chains. *Ann. Appl. Probab.*, 12(1):224–247, 2002. ISSN 1050-5164. URL <https://doi.org/10.1214/aoap/1015961162>.
- Søren F. Jarner and Gareth O. Roberts. Convergence of heavy-tailed Monte Carlo Markov chain algorithms. *Scand. J. Statist.*, 34(4):781–815, 2007. ISSN 0303-6898. doi: 10.1111/j.1467-9469.2007.00557.x. URL <http://dx.doi.org/10.1111/j.1467-9469.2007.00557.x>.
- Søren Fiig Jarner and Ernst Hansen. Geometric ergodicity of Metropolis algorithms. *Stochastic Process. Appl.*, 85(2):341–361, 2000. ISSN 0304-4149. doi: 10.1016/S0304-4149(99)00082-4. URL [http://dx.doi.org/10.1016/S0304-4149\(99\)00082-4](http://dx.doi.org/10.1016/S0304-4149(99)00082-4).
- Galin L. Jones, Murali Haran, Brian S. Caffo, and Ronald Neath. Fixed-width output analysis for Markov chain Monte Carlo. *J. Amer. Statist. Assoc.*, 101(476):1537–1547, 2006. ISSN 0162-1459. URL <https://doi.org/10.1198/016214506000000492>.
- C. Kipnis and S. R. S. Varadhan. Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions. *Comm. Math.*

Phys., 104(1):1–19, 1986. ISSN 0010-3616. URL <http://projecteuclid.org/euclid.cmp/1104114929>.

T. L. Lai and D. Siegmund. A nonlinear renewal theory with applications to sequential analysis. II. *Ann. Statist.*, 7(1):60–76, 1979. ISSN 0090-5364. URL [http://links.jstor.org/sici?sici=0090-5364\(197901\)7:1<60:ANRTWA>2.0.CO;2-L&origin=MSN](http://links.jstor.org/sici?sici=0090-5364(197901)7:1<60:ANRTWA>2.0.CO;2-L&origin=MSN).

Krzysztof Łatuszyński and Jeffrey S. Rosenthal. The containment condition and AdapFail algorithms. *J. Appl. Probab.*, 51(4):1189–1195, 2014. ISSN 0021-9002. URL <https://doi.org/10.1239/jap/1421763335>.

Krzysztof Łatuszyński, Miasojedow, and Wojciech Niemiro. Nonasymptotic bounds on the estimation error of MCMC algorithms. *Bernoulli*, 19(5A):2033–2066, 2013a. ISSN 1350-7265. doi: 10.3150/12-BEJ442. URL <http://dx.doi.org/10.3150/12-BEJ442>.

Krzysztof Łatuszyński, Gareth O. Roberts, and Jeffrey S. Rosenthal. Adaptive Gibbs samplers and related MCMC methods. *Ann. Appl. Probab.*, 23(1):66–98, 2013b. ISSN 1050-5164. doi: 10.1214/11-AAP806. URL <http://dx.doi.org/10.1214/11-AAP806>.

Jun S. Liu. *Monte Carlo strategies in scientific computing*. Springer Series in Statistics. Springer-Verlag, New York, 2001. ISBN 0-387-95230-6.

David J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, New York, 2003. ISBN 0-521-64298-1.

Tristan Marshall and Gareth Roberts. An adaptive approach to Langevin MCMC. *Stat. Comput.*, 22(5):1041–1057, 2012. ISSN 0960-3174. URL <https://doi.org/10.1007/s11222-011-9276-6>.

Nicholas Metropolis and S Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953. ISSN 00219606. doi: <http://dx.doi.org/10.1063/1.1699114>. URL http://jcp.aip.org/resource/1/jcpsa6/v21/i6/p1087/_s1?bypassSS0=1.

- Sean Meyn and Richard L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, Cambridge, second edition, 2009. ISBN 978-0-521-73182-9. doi: 10.1017/CBO9780511626630. URL <http://dx.doi.org/10.1017/CBO9780511626630>. With a prologue by Peter W. Glynn.
- Marian Mureşan. *A concrete approach to classical analysis*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2009. ISBN 978-0-387-78932-3. doi: 10.1007/978-0-387-78933-0. URL <http://dx.doi.org/10.1007/978-0-387-78933-0>.
- Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov chain Monte Carlo*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, pages 113–162. CRC Press, Boca Raton, FL, 2011.
- David J. Nott and Robert Kohn. Adaptive sampling for Bayesian variable selection. *Biometrika*, 92(4):747–763, 2005. ISSN 0006-3444. URL <https://doi.org/10.1093/biomet/92.4.747>.
- Esa Nummelin. Mc’s for mcmc’ists. *International Statistical Review*, 70(2):215–240, 2002. ISSN 1751-5823. doi: 10.1111/j.1751-5823.2002.tb00361.x. URL <http://dx.doi.org/10.1111/j.1751-5823.2002.tb00361.x>.
- M. Ottobre, N. S. Pillai, and K. Spiliopoulos. Optimal Scaling of the MALA algorithm with Irreversible Proposals for Gaussian targets. *ArXiv e-prints*, February 2017.
- Michael L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM J. Matrix Anal. Appl.*, 9(2):256–268, 1988. ISSN 0895-4798. doi: 10.1137/0609021. URL <http://dx.doi.org/10.1137/0609021>. SIAM Conference on Linear Algebra in Signals, Systems, and Control (Boston, Mass., 1986).
- Michael L. Overton. Large-scale optimization of eigenvalues. *SIAM J. Optim.*, 2(1):88–120, 1992. ISSN 1052-6234. URL <https://doi.org/10.1137/0802007>.
- Natesh S. Pillai, Andrew M. Stuart, and Alexandre H. Thiéry. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *Ann. Appl. Probab.*, 22(6):2320–2356, 2012. ISSN 1050-5164. URL <https://doi.org/10.1214/11-AAP828>.
- M. Pollock, P. Fearnhead, A. M. Johansen, and G. O. Roberts. The Scalable Langevin Exact Algorithm: Bayesian Inference for Big Data. *ArXiv e-prints*, September 2016.

- Michael Reed and Barry Simon. *Methods of modern mathematical physics. I*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York, second edition, 1980. ISBN 0-12-585050-6. Functional analysis.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22:400–407, 1951. ISSN 0003-4851.
- Christian Robert and George Casella. A short history of Markov chain Monte Carlo: subjective recollections from incomplete data. *Statist. Sci.*, 26(1):102–115, 2011. ISSN 0883-4237. URL <https://doi.org/10.1214/10-STS351>.
- Christian P. Robert and George Casella. *Monte Carlo statistical methods*. Springer Texts in Statistics. Springer-Verlag, New York, second edition, 2004. ISBN 0-387-21239-6. URL <https://doi.org/10.1007/978-1-4757-4145-2>.
- G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. Roy. Statist. Soc. Ser. B*, 59(2): 291–317, 1997a. ISSN 0035-9246. doi: 10.1111/1467-9868.00070. URL <http://dx.doi.org/10.1111/1467-9868.00070>.
- G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. Roy. Statist. Soc. Ser. B*, 59(2): 291–317, 1997b. ISSN 0035-9246. URL <https://doi.org/10.1111/1467-9868.00070>.
- G. O. Roberts and A. F. M. Smith. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Process. Appl.*, 49(2):207–216, 1994. ISSN 0304-4149. URL [https://doi.org/10.1016/0304-4149\(94\)90134-1](https://doi.org/10.1016/0304-4149(94)90134-1).
- G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodol. Comput. Appl. Probab.*, 4(4):337–357 (2003), 2002. ISSN 1387-5841. URL <https://doi.org/10.1023/A:1023562417138>. International Workshop in Applied Probability (Caracas, 2002).
- G. O. Roberts and R. L. Tweedie. Bounds on regeneration times and convergence rates for Markov chains. *Stochastic Process. Appl.*, 80(2):211–229, 1999. ISSN 0304-4149. doi: 10.1016/S0304-4149(98)00085-4. URL [http://dx.doi.org/10.1016/S0304-4149\(98\)00085-4](http://dx.doi.org/10.1016/S0304-4149(98)00085-4).
- G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, 7(1):110–120, 1997.

ISSN 1050-5164. doi: 10.1214/aoap/1034625254. URL <http://dx.doi.org/10.1214/aoap/1034625254>.

Gareth O. Roberts and Jeffrey S. Rosenthal. Geometric ergodicity and hybrid Markov chains. *Electron. Comm. Probab.*, 2:no. 2, 13–25 (electronic), 1997. ISSN 1083-589X. doi: 10.1214/ECP.v2-981. URL <http://dx.doi.org/10.1214/ECP.v2-981>.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 60(1):255–268, 1998. ISSN 1369-7412. URL <https://doi.org/10.1111/1467-9868.00123>.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statist. Sci.*, 16(4):351–367, 2001. ISSN 0883-4237. doi: 10.1214/ss/1015346320. URL <http://dx.doi.org/10.1214/ss/1015346320>.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probab. Surv.*, 1:20–71, 2004. ISSN 1549-5787. doi: 10.1214/154957804100000024. URL <http://dx.doi.org/10.1214/154957804100000024>.

Gareth O. Roberts and Jeffrey S. Rosenthal. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Probab.*, 44(2):458–475, 2007. ISSN 0021-9002. doi: 10.1239/jap/1183667414. URL <http://dx.doi.org/10.1239/jap/1183667414>.

Gareth O. Roberts and Jeffrey S. Rosenthal. Examples of adaptive MCMC. *J. Comput. Graph. Statist.*, 18(2):349–367, 2009. ISSN 1061-8600. URL <https://doi.org/10.1198/jcgs.2009.06134>.

Gareth O. Roberts and Richard L. Tweedie. Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):341, 1996. ISSN 13507265. doi: 10.2307/3318418. URL <http://www.jstor.org/stable/3318418?origin=crossref>.

Gareth O. Roberts and Richard L. Tweedie. Geometric L^2 and L^1 convergence are equivalent for reversible Markov chains. *J. Appl. Probab.*, 38A:37–41, 2001. ISSN 0021-9002. doi: 10.1239/jap/1085496589. URL <http://dx.doi.org/10.1239/jap/1085496589>. Probability, statistics and seismology.

Steven Roman. *The umbral calculus*, volume 111 of *Pure and Applied Mathematics*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York, 1984. ISBN 0-12-594380-6.

- Jeffrey S. Rosenthal. AMCMC: An R interface for adaptive MCMC. *Comput. Statist. Data Anal.*, 51(12):5467–5470, 2007. ISSN 0167-9473. URL <https://doi.org/10.1016/j.csda.2007.02.021>.
- Jeffrey S. Rosenthal. Optimal proposal distributions and adaptive MCMC. pages 93–111, 2011.
- Eero Saksman and Matti Vihola. On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. *Ann. Appl. Probab.*, 20(6):2178–2203, 2010. ISSN 1050-5164. doi: 10.1214/10-AAP682. URL <http://dx.doi.org/10.1214/10-AAP682>.
- D. Sejdinovic, H. Strathmann, M. L. Garcia, C. Andrieu, and A. Gretton. Kernel Adaptive Metropolis-Hastings. *ICML*, pages 1665–1673, 2014. URL <http://arxiv.org/abs/1307.5302>.
- Antti Solonen, Pirkka Ollinaho, Marko Laine, Heikki Haario, Johanna Tamminen, and Heikki Järvinen. Efficient MCMC for climate model parameter estimation: parallel adaptive chains and early rejection. *Bayesian Anal.*, 7(3):715–736, 2012. ISSN 1936-0975. URL <https://doi.org/10.1214/12-BA724>.
- Michael Spivak. *Calculus*. Publish or Perish, 3 edition, 1994. ISBN 978-0914098898. doi: 10.4135/9781412983556.
- O. Stramer and R. L. Tweedie. Langevin-type models. I. Diffusions with given stationary distributions and their discretizations. *Methodol. Comput. Appl. Probab.*, 1(3):283–306, 1999. ISSN 1387-5841. URL <https://doi.org/10.1023/A:1010086427957>.
- Luke Tierney. A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.*, 8(1):1–9, 1998. ISSN 1050-5164. URL <https://doi.org/10.1214/aoap/1027961031>.
- A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 2000.
- Matti Vihola. On the stability and ergodicity of adaptive scaling Metropolis algorithms. *Stochastic Process. Appl.*, 121(12):2839–2860, 2011. ISSN 0304-4149. doi: 10.1016/j.spa.2011.08.006. URL <http://dx.doi.org/10.1016/j.spa.2011.08.006>.
- Matti Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Stat. Comput.*, 22(5):997–1008, 2012. ISSN 0960-3174. doi: 10.1007/s11222-011-9269-5. URL <http://dx.doi.org/10.1007/s11222-011-9269-5>.

- W. Wang and M. Á. Carreira-Perpiñán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *ArXiv e-prints*, September 2013.
- C. Zălinescu. *Convex analysis in general vector spaces*. World Scientific Publishing Co., Inc., River Edge, NJ, 2002. ISBN 981-238-067-1. URL <https://doi.org/10.1142/9789812777096>.